Optimization and Application of a Flexible Dual Arm Robot Based Automation System for Sample Preparation and Measurement

Dissertation to obtain the academic degree of

Doctor of Engineering (Dr.-Ing.)

from the Faculty of Computer Science and Electrical Engineering at

University of Rostock

Submitted by:

Shalaka Joshi, born on 19. October 1990 in Pune,

India From Pune, India

Rostock, Germany, 25th March 2021

https://doi.org/10.18453/rosdok_id00003453

Reviewers:
Prof. Dr.-Ing. habil. Kerstin Thurow, University of Rostock, Rostock.
Prof. Dr.-Ing. habil. Heidi Fleischer, University of Rostock, Rostock.
Prof. John Wen, Rennselaer Polytechnic Institute-Troy, New York.

Year of submission: 2021
Year of defense : 2021

Acknowledgements

The completion of this thesis could not have been possible without the assistance of a number of individuals.

First and for most, I wish to express my gratitude to my supervisors Prof. Dr.-Ing. habil. Kerstin Thurow and Prof. Dr.-Ing. habil. Heidi Fleischer for their elaborate guidance, continuous support and precious knowledge. Their suggestions inspired me in all the time of researching and writing of this dissertation. I would especially like to thank Dr.-Ing. Thomas Roddelkopf, Dipl.-Ing. Lars Woinar, Heiko Engelhardt, Sybille Horn, B. Sc. Anne Reichelt and Dr.-Ing. Steffen Junginger for their competent help.

I would also like to thank all my colleagues of the Institute of Automation and the Center for Life Science Automation (Celisca) at University of Rostock, Germany. This dissertation would not be successful without their inputs and help.

I would like to acknowledge the timely support of Dr.-Ing. Michael Klos, B.Eng. Wolfgang Schuberthan and Gerhard Trapp of Yaskawa Europe GmbH.

Table of Contents

1 Introduction 1

2 Introduction to Dual-Arm Robots 4

2.1 Dual-arm Robots in Laboratory Applications 5

2.1.1 Motoman Dual Arm Robot 5

2.1.2 Dual-arm Robotic System for Flexible Screening 8

2.1.3 RAMSAY-2 Dual-arm Robot 9

2.1.4 ABB YuMi Cobot 10

2.1.5 Andrew Dual-Arm Robots 11

2.1.6 Other Dual-arm Robots Used in Laboratory Applications 13

2.2 Motion Control of Dual-arm Robots 16

2.2.1 C-space Mapping Using the A*, RRT* Algorithms 16

2.2.2 Motion Planning Based on Gesture Recognition 17

2.2.3 Constraint Based Task Programming 19

2.2.4 Fuzzy Control of Dual-arm Robots 21

2.3 Summary 22

3 System Overview 23

3.1 Robot Hardware 23

3.2 Labware and Instruments 26

3.3 SAMI Ex User Interface 31

3.4 Robot Interface (R-Interface) 37

4 Aims of Dissertation 39

5 Optimization Strategies 42

5.1 Motion Planning Using Teach Pendant 42

5.2 Path Planning Algorithms 44

5.3 Offline Motion Planning....45
5.4 Comparison of Possible Approaches....47
5.5 Final Approach....48
6 New Technology Introductions....52
6.1 Electronic Pipettes....52
6.2 Gripper System....55
6.3 Positive Pressure Unit for Solid Phase Extraction (SPE)....58
6.4 Glass Syringe....58
6.5 Summary....60
7 Basics of Motion Elements....61
7.1 Robot Posture....61
7.1.1 Path Motion Element....62
7.1.2 Relative Motion Element....63
7.2 Reference Points for Motion Elements....64
8 Generation of Motion Elements....67
8.1 Standby Position and Intermediate Positions....67
8.2 Node Points and Key Points....68
8.3 Naming of a Motion Element....71
8.4 Designing a Motion Element....73
9 Flexible Motion Planning....78
9.1 Combining Motion Elements....78
9.1.1 Job Search Algorithm and Directory Refresh....78
9.1.2 File Transfer Functionality....82
9.1.3 Error Handling in Remote Mode....83
9.2 Motion Planning Algorithm....84

9.3 SAMI Method Creation 89
10 Motion Elements in Applications 92
10.1 Pipette 93
10.2 Electronic Pipette 95
10.3 Glass Pipette 97
10.4 Shelf transfer 98
10.5 Hotel transfer 100
10.6 Syringe 101
10.7 Glass Syringe 102
10.8 Vial Task 103
10.9 Ultrasonic Shaker and Shaker Task 106
10.10 Liquid Chromatography 107
10.11 Gas Chromatography 109
10.12 Thermoshaker 110
10.13 Solid Phase Extraction 111
11 Realized Applications 113
11.1 Application I: Sample Preparation for Detection of Chiral Enantiomers 113
11.1.1 Description of Process 114
11.1.2 Results and Discussion 116
11.1.3 Batch Sample Preparation 119
11.2 Application II: Detection of cholesterol levels in biliary stent samples 121
11.2.1 Description of Process 121
11.2.2 Results and Observations 123
11.2.3 Batch Processing of samples 126
12 Conclusion and Future Scope 129

13 Bibliography 133

14 Appendix 144

14.1 Repetitions and success/failure rate for each task 144

14.1.1 Pipetting with Multi-Channel Pipettes 144

14.1.2 Single- Channel Pipettes 147

14.1.3 Pipetting with electronic pipettes 151

14.1.4 Vial Task 154

14.1.5 Syringe Liquid Transfer 161

14.1.6 Glass Syringe Liquid Transfer 166

14.1.7 Shaker/Ultrasonic task 169

14.1.8 Thermoshaker Task 172

14.2 List of Variables BXXX used 175

14.3 List of PXXX Variables used 177

14.4 List of BPXXX Variables 179

14.5 Settings for each Task in SAMI Ex Run Robot Box 180

14.6 List of Motion Elements 181

14.7 Checklist and Tips for Robot Job Generation 185

Declaration 186

List of Publications 187

Theses 188

Abstract 190

Zusammenfassung 192

List of Figures

Figure 1: Motoman SDA20D Dual-arm Robot [42] 5

Figure 2: Dupalro anti-cancer drug compounding system employing the Yaskawa Motoman dual-arm robot [47] 6

Figure 3: Yaskawa Motoman robot implemented for analysis of reconstructed human epidermis [48] 7

Figure 4: Stäubli RX90 robots for flexible screening processes [49] 8

Figure 5: RAMSAY-2 robot for processing of biological samples [19] 9

Figure 6: ABB Yumi Cobot with dual arms [65] 10

Figure 7: ABB's YUMI laboratory and hospital cobot [65] 11

Figure 8: (a): Andrew pipetting robot with volume adjusting jig for manual pipettes, (b): Andrew+ robot for electronic pipettes [73] 12

Figure 9: (a): Kawasaki duAro dual-arm SCARA robot [79], (b): Yamaha dual-arm cartesian robot [80] 13

Figure 10: Tecan Cavro Omni dual-arm cartesian robot for automatio of preparation of fungal sample [81] 14

Figure 11: Xantus dual-arm cartesian robot for in-meso crystallization of membrane proteins [82] 15

Figure 12: (a): SCARA dual-arm robot path planning using A* heuristics, (b): C-space discretization. S is the start point, G is the target point, u1 and u2 are middle points.Q0,Q1,Q2,Q3 are the obstacle boundaries with robot arms at positions S, G,u1 and u2 [74]. 16

Figure 13: Dual arm robot attempting to pick a cup from the table using the rapidly exploring random tree (RRT) algorithm. The image shows all intermediate positions of the arms while traversing from under the table to the target object [95]. 17

Figure 14: System architecture of the controlling system to direct a Yaskawa Motoman robot using gestures recognition [100] 18

Figure 15: Standard process flow of detecting, grasping and gesture directed transportation of an object [100]. 19

Figure 16: Dual-arm robot being taught using gesture recognition [12] 19

Figure 17: Dual-arm robot creating cell cultures with synchronous arm movements controlled via fuzzy logic [104] 20

Figure 18: Fuzzy logic controller for the motion planning of a dual-arm robot in the absence of robot dynamic and kinematic information [107] 21

Figure 19: Yaskawa SDA10F reachability envelope [42] 23

Figure 20: 3D rendering of the robot environment. (a) Safety door, (b) Infrared light curtain, (c) Manual pipette, (d) Electronic pipette, (e) Dual-arm robot, (f) Gas Chromatography Mass Spectrometer (GC/MS), (g) Liquid Chromatography Mass Spectrometer (LCMS), (h)Workbench with instruments, (i) Horizontal shelves, (j) Vertical storage, (k) Solid Phase Extraction (SPE) 24

Figure 21: Yaskawa SDA10F dual-arm robot environment with instruments; (a) LC/MS, (b) GC/MS, (c) Manual pipettes, (d) Thermoshaker, (e) Workbench, (f) Cooling rack, (g) LCMS tray, (h) Shaker, (i) Ultrasonic bath, (j) Temporary pipette stand (k) Vertical storage, (l) Horizontal storage, (m) Electronic pipettes, (n) Dual-arm robot, (o) SPE 25

Figure 22: Top view of robot enclosure and relative positions of laboratory instruments in RS-2 25

Figure 23: ALPs on the workbench 26

Figure 24: Manual pipettes hanging on a rack with the help of adapters for easy robot grasp. 27

Figure 25: Pipettes hung on a temporary pipette stand 27

Figure 26: Various tip boxes in special adaptors 28

Figure 27: (a) Vertical Storage, (b)Horizontal labware storage 28

Figure 28:(a) & (b) 3D printed vial and lid holder, (c) Different sizes of vials 29

Figure 29: Eight tank reservoir and 96 well microtiter plate for sample preparation 29

Figure 30: (a): Syringes, (b): Filters, (c): Cannulas, (d): Glass pipettes 30

Figure 31: (a): Shaker, (b): Thermoshaker, (c): Cooling rack, (d): Ultrasonic bath 30

Figure 32: System Configuration 31

Figure 33: Architecture of the process automation system 32

Figure 34: New labware definition for importing into SAMI Ex methods 33

Figure 35: Depicts the specifications of a pipetting task using manual pipettes. The 'Volume' and 'Pipette' fields are used to identify the pipette type, 'Channel' field differentiates between the single and multi-channel pipettes, 'PrePipette' field is set if pre-wetting of the pipetting tip is required.....34

Figure 36:SAMI Ex Method Scheduler for process verification.....35

Figure 37: Process run of an ultrasonic shaker task in SAMI Runtime software. The process status can be observed remotely in real-time by tracking the motion of labware icons. 36

Figure 38: Block representation of robot interface architecture 37

Figure 39: XML communication between the SAMI Ex and R-Interface for a vial transfer task between the ultrasonic shaker and mechanical shaker.....38

Figure 40: Previous interface and job structure where the robot tasks always start in the standby position.....39

Figure 41: API for moving the robot arm directly via the interface program.....43

Figure 42: Utilizing robot kinematics to compute the motion without a fixed start and end point for a task.....44

Figure 43: Classification of path planning algorithms.....44

Figure 44: High level diagram of a robot controlled using ROS framework in collaboration with compatible third-party open-source software 46

Figure 45: Overview of the communication paths within ROS framework 47

Figure 46: Example job for loading tips to pipette 50

Figure 47: Proposed process flow for flexible and fast sample preparation.....51

Figure 48: (a) Electronic pipettes hanging on the overhead shelf, (b): single arm operation of pipetting liquid into a vial 53

Figure 49: (a) available COM ports, (b) query pipette properties, (c) initialize pipette (d) set speed, (e) set piston position, (f) check tip eject, (g) set volume of liquid, (h) command sent and reply received, ...54

Figure 50: Previous end effectors (grippers) without force sensing 55

Figure 51: Width difference in MTP and MTP lid requiring different grip settings 55

Figure 52: New gripper fingers with force sensing 56

Figure 53: Example of motion element implementing the new force feedback grippers 57

Figure 54:(a): SPE instrument with (1) SPE hood, (2) cartridge, (3) rinsing block, (4) carriage, (b) Specially designed 2 mL vial block with capacity for 24 vials, (c) cartridge adaptor, (d) rinsing block.....58

Figure 55: Glass syringe mounted into the holder (a): adaptor prototype during syringe change, (b): final adaptor design during syringe change 59

Figure 56: Glass syringe actuated by the robot; (a): fingers of the left arm actuating the plunger, (b): robot arm posture with the cooling rack placed on the ALP.....60

Figure 57: Adjusting the arm posture for maximum reachability, (a) correct position to reach all the ALPs on the workbench, (b) arm posture limits the reachability of the end effector to only a few ALPs.....62

Figure 58: Types of motion elements 62

Figure 59: Job content depicting highlighting the robot joint angles for both the arms saved in a path motion.....63

Figure 60: Graphic depicting the algorithm for a relative motion element 64

Figure 61: Reference points (a) Hotel 0 room 0 as hotel reference point, (b) shelf position 0 as shelf reference point.....65

Figure 62: Reference point for each ALP according to the existing system [111].....65

Figure 63: Reference points on ALPs for (a) right arm, (b) left arm.....66

Figure 64: Standby Position.....67

Figure 65: Intermediate Position.....67

Figure 66: Node classification 68

Figure 67: (a) pick vial motion element with different start and node points in RS-1, (b) put vial motion element with different start and end node points in RS-1, (c): single motion element with identical start and end node points for picking and placing vials in RS-2.....69

Figure 68: (a) Hotel key point, (b) workbench key point, (c) shelf key point, (d) pipette key point 70

Figure 69: Task design, (a): All tasks starting and ending at the standby position in RS-1, (b): Tasks starting and ending at one of the key points for continuous and flexible task planning in RS-2: 71

Figure 70: Subdivision of a task into five primary motion components.....73

Figure 71: List of some motion elements for transferring labware in and out of the hotels as seen in RS-1 74

Figure 72: Graphic showing an example of a motion element for aspirating liquid from multiple labware cases using a manual pipette 76

Figure 73: Example of a hotel task job calling the various hotel motion elements in sequence and setting the required variables 78

Figure 74: Table depicting keywords and motion element translations included in old system design 79

Figure 75: Algorithm for finding consecutive motion elements in a task according to RS-1 .80

Figure 76: Algorithm for finding motion elements in a task according RS-2 82

Figure 77: Flowchart for sending motion element to robot controller automatically in RS-2 83

Figure 78: Motion planning algorithm in RS-1 85

Figure 79: Task flow of an example process in RS-1. The robot goes to the standby position after each task 85

Figure 80: Motion planning algorithm for connected tasks in RS-2 87

Figure 81: Task flow of an example process following the path planning approach in RS-2. The robot goes to the intermediate node only when necessary. 87

Figure 82: Robot arm position in (a) workbench key point (b) pipette key point 88

Figure 83: Error message by the SAMI Runtime software for duplicated labware positions in a multi-method queue 89

Figure 84: Method file in SAMI Ex software for sample preparation process for determination of cholesterol in biliary stent incrustations. 90

Figure 85: Three families of vials placed on the ALPs. The three families cannot be placed on one ALP. 90

Figure 86: Runtime system view of a running process depicting the various labware and their positions, the family number being run and the ongoing task in real time. 91

Figure 87: Template for understanding the motion element diagrams detailing the flow of labware, arms used, modes reached and the motions carried out. 93

Figure 88: Pipetting task (a) pick/ put back pipette, (b) load tips, (c) aspirate/dispense liquid, (d) reject tip 94

Figure 89: Motion elements for manual pipetting tasks (a): according to RS-1 [111], (b):according to RS-2.....94

Figure 90: Electronic pipettes. (a): pick / place pipette, (b): load tip, (c): aspirate/dispense liquid, (d): reject tip 95

Figure 91: Figure depicting the motion elements for the electronic pipette task in RS-2 96

Figure 92: Glass pipette: (a) prepare to pick pipette (b) pick pipette, (c) aspirate liquid, (d) dispense liquid, (e) put pipette, (f) reject pipette 97

Figure 93: Figure depicting the motion elements for glass pipette task (a):according to RS.1 [111] (b): according to RS-2 97

Figure 94: Transfer from shelf: (a) pick labware, (b) place on elevated ALP, (c) change grip position, (d) place labware on workbench. 98

Figure 95: Motion elements for labware transfer between shelf and workbench (a according to RS.1 [111] (b): according to RS-2 99

Figure 96: Hotel transfer (a): Pick labware, (b): place on grip changing station, (c): change grip, (d): place on workbench.....100

Figure 97: Motion elements for transferring labware from vertical storage to workbench; (a): according to RS.1 [111] (b): according to RS-2 100

Figure 98: Syringe task (a): pick syringe from rack, (b): transfer to arm, (c): pick cannula, (d): aspirate liquid, (e): reject cannula, (f): pick filter, (g): dispense liquid, (h): reject syringe and filter.....101

Figure 99: Motion elements for the syringe task; (a): according to RS.1 [111] (b): according to RS-2 102

Figure 100: Motion elements for liquid transfer using glass syringe in RS-2 103

Figure 101: Vial opening (a): approach vial, (b): pick vial, (c): transfer vial to other arm, (d): unlid vial, (e): place vial lid 104

Figure 102: Motion elements for vial task; (a): according to RS.1 [111] (b): according to RS-2.....104

Figure 103: Ultrasonic bath and shaker task: (a) pick vial, (b) transfer vial, (c) place vial in ultrasonic bath / shaker instrument 106

Figure 104: Figure depicting the motion elements for mixing task with shaker and ultrasonic bath; (a): according to RS.1 [111] (b): according to RS-2.....106

Figure 105: LC task (a) pick labware with arm 1, (b) transfer labware, (c) place labware on tray with arm 1, (d) transfer tray with arm 2, (e) place tray in LC with arm 2 107

Figure 106: Motion elements for LC-autosampler tasks; (a): according to RS.1 [111] (b): according to RS-2 108

Figure 107: GC task (a): pick vial, (b) transfer vial to arm 1, (c) transfer vial to arm 2, (d)place vial to GC autosampler 109

Figure 108: Wells marked by red programmed for RS-1, wells marked by green programmed for RS-2 109

Figure 109: Motion elements for the task of loading sample vials to the GC autosampler, (a): according to RS.1 [111] (b): according to RS-2 110

Figure 110: Thermoshaker task (a) open thermoshaker with arm 2, (b)pick labware from workbench with arm 1, (c): transfer labware, (d) place labware to thermoshaker, (e) close thermoshaker 110

Figure 111: Motion elements for thermoshaker task; (a) according to RS.1 [111] (b): according to RS-2 111

Figure 112: SPE task (a): pick rinsing block from hotel, (b) place labware on SPE, (c) pick cartridge from shelf, (d) stack cartridge on previous labware 112

Figure 113: Motion elements for SPE task in RS-2 112

Figure 114: Block diagram depicting the steps of sample preparation for detecting chiral enantiomers 114

Figure 115: Flowchart depicting the chiral sample preparation process. 114

Figure 116: Graphical comparison of the calibration measurements prepared manually and using the dual-arm robot 117

Figure 117: Graph depicting the average intensity ratios of unknown samples: 117

Figure 118: Labware placement on workbench during sample preparation 119

Figure 119: Process flow of cholesterol sample preparation 122

Figure 120: Workbench layout for the cholesterol application 123

Figure 121: Comparison between manual and robotically prepared calibration solutions for cholesterol analysis 124

List of Tables

Table 1: Commands processed by the GC-MS interface, LC-MS interface and the R-Interface 33

Table 2: List of labware descriptions as seen in the Biomek software 34

Table 3: Qualitative comparison of the three path planning approaches 48

Table 4: Depiction of the various liquid volumes dispensed using a combination of four single channel manual pipettes set to a fixed value 52

Table 5: Commands and actions implemented by the electronic pipettes 53

Table 6: Right and left end effector grip widths for various labware 55

Table 7: Table depicting the rotation limits of each robot arm joint 61

Table 8: Key points and corresponding tasks that can be carried out without moving the robot to the intermediate position 70

Table 9: Table representing in short the changes to planning of motion elements 75

Table 10: List of motion elements for a manual pipette tasks required in RS-1 and RS-2 75

Table 11: Example of directory with motion elements listed against the corresponding code name 81

Table 12: List of all the tasks performed by the robot 92

Table 13: LC Measured values of both enantiomers in manually prepared and robot prepared samples 116

Table 14: Speed comparison between the two automated methods. 118

Table 15: List and quantity of labware for four batches of samples for analysis of chirality of compounds 119

Table 16: List of all the motions required for chiral sample preparation with success / failure for each well column in MTP 120

Table 17: Table depicting the accuracy of calibration samples as detected by the GC 123

Table 18: Sample repeatability 124

Table 19: Table depicting the analysis results pertaining to the concentration of cholesterol in samples as measured by the GC 125

Table 20: Measurement values for samples with 5-a-cholestane mixed in stock solution....125

Table 21: Time required by the robot in RS-2 to produce 12 samples.....125

Table 22: List of all labware required for running four batches of sample preparation with position of each labware before/after and during the run 126

Table 23: Table depicting the motions used for sample preparation to detect 127

List of Abbreviations

LC/MS: Liquid chromatography mass spectrometry

GC/MS: Gas Chromatography Mass Spectrometry

MTP: Microtiter plate

TM: Thermoshaker

PE: Air displacement pipette

EP: Electronic pipette

GP: Glass pipette

GS: Glass syringe

US: Ultrasonic bath

CR: Cooling rack

TSIM: 1-(Trimethylsilyl) imidazole

SPE: Solid phase extraction

VI: Vials

ALP: Automated labware placement

Definitions of Commonly Used Terms

1. Labware: All the 3D printed adaptors, vials, microplates, pipettes are called labware.

2. Hotels: Custom made vertical storage structures built on the workbench to utilize vertical space for storing labware like microplates and reservoirs.

3. Shelf: Custom made horizontal storage structures built above the workbench to store frequently used labware like tip boxes.

4. ALP: Automated labware placement structures built on the workbench to place individual labware during sample preparation.

5. Motion elements: Short movements of the arm to reach from one position to the other. They are also called ‘jobs’ or ‘robot jobs’.

6. Task: Individual chores done by the robot e.g.: pipetting, transportation, incubation in thermoshaker, vial opening, etc. A task is built using a series of motion elements.

7. Process: Sample preparation application implemented on the robot. A process is created by running a series of tasks.

8. Method: The graphical interpretation of a process as created in the SAMI Ex software.

9. Node: Start and end positions of a motion element.

10. Keypoints: Start and end postures of a task. Four selected postures of the robot arm where multiple tasks can be connected to be carried out in continuation forming a process.

11. Reference points: Defined points in the robot environment used for creating motion frames.

12. Motion frames: Local coordinates to which a universal motion is mapped

1 Introduction

Automation systems and robots have entered the daily workings in nearly all areas of life. It is the technology by which various tasks and procedures can be done with minimal human interaction and involvement. The role of such systems is to reduce the effort involved in monotonous and repetitive manual jobs [1],[2]. Such systems can routinely be seen in industrial manufacturing applications along with a steady increase in medical and life science applications, bionics and drug delivery. Automated systems were initially developed with standalone machines responsible for performing a certain predefined task. Technology has developed ever since and the scope for mechanized assistance has expanded manifold to allow a closer collaboration between machines and humans. The core of such systems is reliable and fast transfer of information. Robots play an important role in automating any facility as no production facility is without transfer of materials, which the robot can perform in a number of ways. Dual-arm robots are the next upcoming trend to perform tasks in various environments and were primarily developed to assist processes in manufacturing industries [3]. The dual arms provide a human like dexterity and such robotic systems can be programmed to replace the presence of humans in stressful and harmful work environment. It alleviates plant workers from mundane menial jobs to invest time and effort in developing processes and technology.

Dual-arm robots have traditionally been used for industrial applications and for handling large and heavy objects. They might lack the accuracy required in low volume applications in a laboratory setup and may not be suitable to handle the fragile lab equipment. Their advent in the life science research industry is relatively new and the scope of their application is as yet not fully explored.

Life science applications involve handling of organic or toxic substances, cell cultures etc.; the materials used are extremely expensive. The processes vary frequently and requirements within a process also vary often. A flexible automation system is required that can adapt to the changing scenario. A wide range of tasks should be performed by the robot while being capable of manipulating fragile and sensitive laboratory equipment. The robot is utilized to replicate scientific processes. These processes are meticulously designed to get optimum output and the best results. There is no scope for alteration of the processes. The reagent volumes are sensitive to minute change as very low quantities in microliters are used. Hence, the robot should perform the exact process over and over again.

A dual-arm robot is installed in a laboratory setup and is programmed to perform sample preparation tasks. This thesis describes the optimization of the existing robotic setup. The industrial size robot is programmed to handle all the laboratory equipment used by technicians to carry out processes manually. The ease of programming and adapting the system to various applications and user requirements is of utmost importance. It is also expected to have a high accuracy and repeatability in order to handle microliter volumes. In addition to this the resulting system is also expected to be comparable in speed with a human worker. The thesis focusses in optimization of the in-house developed automation system in order to reduce the time

required for sample preparation processes and introduce flexibility in system design to allow quick and easy system reconfiguration for new processes as well as the processing of multiple batches of samples. The existing motions of the robot are studied. New optimized motions are created to perform the same tasks in an attempt to make the process faster. The optimized system enables continuous preparation of laboratory test samples in multiple batches. The process design is changed to make the system versatile to adapt to a range of applications. The new optimized motion and system design is compared with the initial existing system and the advantages are observed based on the results of the actual samples prepared. The system is expected to be applied in commercial applications with high throughput rates.

Chapter two discusses the current state of art in the automation industry. Application of industrial dual-arm robot in life science automation are discussed with specific examples

The 3rd chapter gives an overview of the robot hardware and software. Details regarding the controller, robot geometry, software architecture and the process scheduling software are included in this chapter.

The fourth chapter elaborates the problem statement for the thesis in detail. Various factors are identified in order to improve the previously installed system. This chapter discusses the grounds of improvement and the various constraints to be considered.

Chapter 5 discusses the various approaches possible to solve the problem at hand. Advantages and disadvantages of all the approaches are discussed. The optimal method and explanation of the method are included in this chapter.

The 6th chapter discusses the implementation of new instruments. The chapter discusses the advantages of the new technology along with challenges in integrating them with the system.

Chapter 7 discusses the basics of motion elements. This chapter highlights the changes done to the reference point system that affect the creation of motion frames. The advantages and limitations of reference points in the original robotic system (RS-1) and the optimized robotic system (RS-2) are discussed.

The 8th chapter describes the process of motion element generation. The chapter discusses the various changes made to the node points and robot postures to make the system flexible and easy to adapt for future applications. It also gives a comparison between the RS-1 and RS-2 and evaluates the advantages of implementing the changes.

A number of changes in terms of job database management, job search conditions, variable allocation and management were introduced in the RS-2. Chapter 9 highlights all these changes and discusses their advantages. The newly developed task planning algorithm is described in this chapter. The chapter also discusses the changes made in the usage of the process scheduling software to overcome the limitations observed in RS-1.

A comparison between the robot motions in the RS-1 and RS-2 is done in chapter 10. The changed motions, advantages of the new motions, challenges while creating the motions are recorded in this chapter with the help of schematic diagrams.

Chapter 11 illustrates two test applications of preparing measurement samples for the enantiomeric determination of proteinogenic amino acids and measuring cholesterol in incrustations of biliary endoprostheses. A description of the process is given in detail. The processed sample solutions are analyzed and the results are discussed in this chapter. A comparison between the measurement results from the RS-1 and RS-2 is included.

The final chapter 12 forms the conclusion of the thesis. The various observations and remarks are recorded in this section. In addition, future improvements and possibilities are included in this chapter.

2 Introduction to Dual-Arm Robots

Life science research includes a range of applications such as microbial testing, elemental analysis, medical sample handling, etc. Extensive sample testing is required on a regular basis in hospitals and private laboratories. This and the increased monetary investment in medical research have propelled scientists into taking efforts to come up with faster and more efficient ways to carry out laboratory experiments and analysis. These factors provide the perfect background to introduce automation systems in laboratory and medical facilities [4]–[7].

Laboratory automation is very different than normal industrial automation and is still an upcoming field. Laboratories need to carry out multiple functions and tests on a given day. The systems need to be easily configurable to accommodate a variety of labware and sample volumes [8]. This requirement is quite opposite to that seen in industries where the installed equipment is utilized for a single task for years together [9],[10]–[14]. Test processes and applications might change frequently and hence the changeover time should be short so that the system downtime can be kept at a minimum. The system is also required to handle delicate and expensive equipment. Such systems enable scientists to optimize and capitalize the available resources. Automated systems can be used for performing specific tasks which if done manually generate a bottleneck in the normal daily working of the laboratory [15]–[17].

The field of laboratory diagnostics is changing rapidly and it is difficult to predict the future requirements. Hence, it is necessary that lab automation solutions are able to counter any future changes in assays and liquid volumes [18]–[21] and should be capable of accommodating future expansion. The lack thereof will lead to reducing the throughput of laboratories, turnaround times and testing capacity [22]–[24]. Hence it is necessary to implement solutions which can be easily extended by adding a new module and for additional sample capacity [25].

For this purpose, current automation systems in labs include a number of individual modules such as shakers and incubators for homogenization, aliquoting devices, sample transporters, analyzers for sample measurement, robots, etc. Intelligent networking and cloud storages make the systems intelligent, for example, patient samples can be bar coded robotically and stored in position coded shelves. Automated systems and robots can retrieve the samples with the help of mapping software reducing the effort of laboratory personnel in searching samples for retests in the future [26]. These various instruments and intelligent storages work synchronously and the requirements are based on individual applications [18], [27]–[32]. The instruments usually focus on fulfilling one specific task of the entire laboratory process. Multiple such instruments have to be connected to each other via conveyor belts. In such setups, the labware processing steps are fixed, limiting the possibility of performing multiple processes consecutively. If one of the apparatus or transportation instruments is out of order, the entire setup has to stop [33]. Such specialized lab instruments often require specific labware configurations. Often, the various instruments cannot directly communicate with each other and often require human assistance for labware transfer.

Thus, a number of single and dual arm robots are being introduced in labs. Dual-arm robots in labs introduces new cost saving opportunities by carrying out the sample preparation tasks

of multiple trained laboratory technicians. Personnel can be reduced in sample preparation and can be increased in the research group [34]. A number of labs are adding automated aliquoters to free up technicians. The technicians do not need to do repetitive tasks and suffer injuries. Lab technicians do not have to be exposed to toxic chemicals on a daily basis.

The following section summarizes the various dual arm robots used for different applications in a laboratory setup including liquid transfer, sample preparation, drug generation, etc. The dual arm robots range from multi DOF human like robots to the more conventional cartesian robots. The multiple degrees of freedom heighten the challenges of programming robot motions in an obstacle dense environment. Various strategies for refining robot motions currently implemented in the industry are summarized in the following section

2.1 Dual-arm Robots in Laboratory Applications

The use of multipurpose dual-arm robots and mobile robots is on the rise in laboratory applications. Dual-arm robots have a number of advantages such as the ability to perform tasks similar to a human being due to the multiple jointed arms. They can be programmed for doing a number of tasks such as transportation and sample processing and can perform tasks simultaneously [6]. More DOF give more posture freedom and can be accommodated in small spaces. The robot can be programmed to communicate with a number of lab instruments in any sequence necessary. Automated systems are controlled with complex communication systems and virtual storages. Special algorithms are used to program the optimal paths and schedule for the robots. Robots and multifunctional instruments can take over the complete working of the laboratories [35]. Labware can be transferred between instruments and eventually the sample can be sent to the analytical module for final testing and result derivation [36],[28],[29],[39]. Analytical instruments such as the GC or LC systems can collaborate with the robot [40],[41].

2.1.1 Motoman Dual Arm Robot

The Motoman SDA series robot by Yaskawa Electric Corporation, Fukuoka, Japan as seen in Figure 1, has a high dexterity due to the dual arms and 15 rotational axes. It has 7 rotational axis per arm and an additional DOF around the waist so that the robot can rotate ± 110 deg around itself to reach objects placed at the back. Each of the rotational joints are actuated using servo motors which lead to precise motions. The robot can carry weights up to 20 kg per arm making it suitable for assembly of heavy objects. The robot is floor mounted making it suitable for long term applications. It has a horizontal reachability of 2,590 mm and vertically 1,820 mm with a positional accuracy of ± 0.1 mm. The construction of the robot is compact as all the electric and pneumatic wiring is enclosed in the robot body. The SDA robots comes in two versions SDAXXD and SDAXXF

Figure 1: Motoman SDA20D Dual-arm Robot c

distinguished by the robot's controller. Each robot in the F version has its unique and compact FS100 controller. The D version robots are controlled using the DX100 controllers which can control up to 8 different robots simultaneously. The robot has inbuilt vibration control and collision detection technology. The programming of the robot is done using a hand held teach pendant using the Inform III programming language [42]. The Yaskawa dual-arm robot is popularly used in assembly applications [43], [44].

2.1.1.1 Anti-cancer Drug Compounding using Yaskawa Motoman Robot

The pharmacists engaged in developing anti-cancer drugs are invariably exposed to toxic chemicals required for the drug on a regular basis which cause side effects such as infertility [47],[48]. It was found that creating safety enclosures assist in reducing some of the effects. Anti-cancer mechanisms require complex tasks to be performed with great accuracy. In order to offset the growing demand for anti-cancer drugs and the diminishing number of trained pharmacists, the research team at National Cancer Center in South Korea has developed an automation system called 'Dupalro' (Figure 2) to create anti-cancer dosages [47]. This Dupalro system is not to be confused with its namesake single-arm robot.

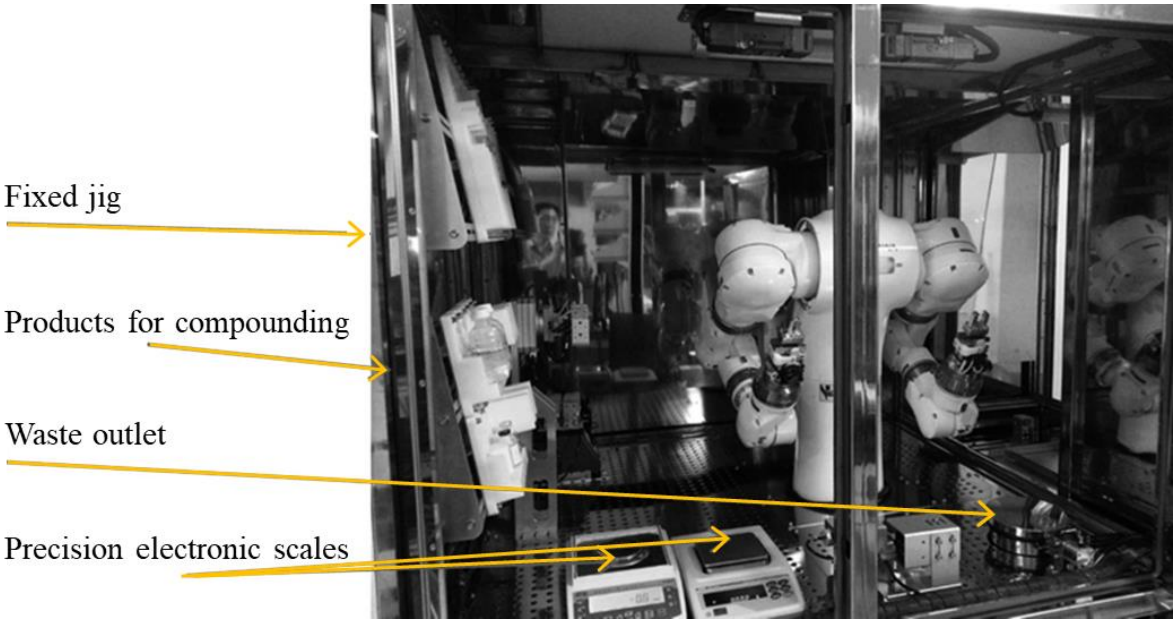


Figure 2: Dupalro anti-cancer drug compounding system employing the Yaskawa Motoman dual-arm robot [47]

The robotic system employs the Yaskawa Motoman dual arm robot. The end effectors were designed to hold various sizes of vials, plastic bags, syringes and to press a plunger. The robot is contained in a custom-made stainless-steel cubicle with specific areas and outlets for product supply, waste discharge and ventilation. The robot receives an EMR (electronic medical record) prescription. The coded reagents required for drug preparation are identified by label readers. The robot checks if all the compounds required for the drugs in the prescription are present on the input tray. The automated preparation starts with a press of a button. A custom-made software was developed which contained the preparation steps for various drugs including the respective compound volumes required for each formulation. The software was programmed to prepare five different drugs. The robot follows individual procedures for the various drugs and mixes the supplied reagents in weighted quantities to prepare different drugs

consecutively in a random order as per prescription. The system is already implemented successfully in Samsung Medical Center, Seoul, South Korea.

2.1.1.2 Automated Downstream Analysis of Epidermal Models

The Yaskawa Motoman SDA10 dual-arm robot has also been implemented at Fraunhofer Institute for Interfacial Engineering and Biotechnology, Wurzburg, Germany. The manual process for the determination of ET-50 value was automated using the robot and the barrier function of a reconstructed human epidermis (RHE) was accessed. ET-50 is the exposure time required for control chemical to reduce cell viability to 50%.

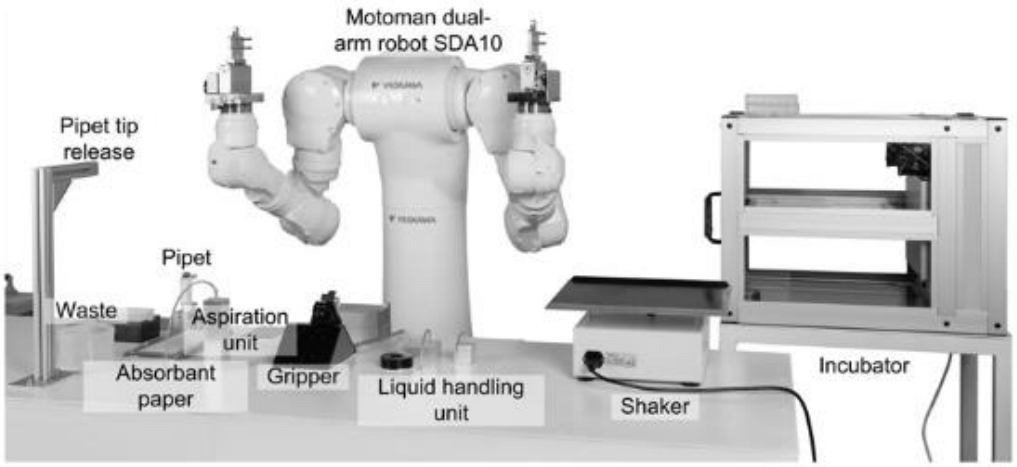


Figure 3: Yaskawa Motoman robot implemented for analysis of reconstructed human epidermis [48]

For the automated process (Figure 3), a library of tasks and working sequence was created on the FS100 controller. Additional standard equipment such as pipettes, pipettes tips, shaker, well plates and cell culture inserts were placed in the immediate vicinity of the robot. The RHE was transferred to a table mounted gripper using a vacuum nozzle on the right robot arm. The solvent application was performed synchronously with both the arms using pipettes. Liquid volumes less than 0.5 mL were transferred using standard laboratory manual pipettes, whereas larger volumes were dispensed using a peristaltic pump. The manual pipettes were operated with both the arms with one arm holding the pipette and the other pushing the piston. In addition, a vacuum pump was installed to assist solvent aspiration during the washing step. The pumps and the robot were controlled by the robot programming unit via digital interfacing. The cell culture plates were then transferred by the robot to an incubator system. The process was carried out simultaneously on 3 RHE samples manually and with the robot. It was seen that the manual process had a better fitting regression equation with a residual sum of squares (RSS) value of 382.3 as compared to the RSS value of 1049.2 for the automated system [48].

2.1.2 Dual-arm Robotic System for Flexible Screening

In order to keep up with the stride in changing drug discovery processes, it is important to develop and maintain a highly flexible and productive screening system. In order to meet the industry demands, MSD (Merck Sharp & Dohme, Kenilworth, USA) in association with RTS Life Science, Manchester, UK have developed and installed a dual-arm robotic screening system (Figure 4) to support early stage drug discovery [49].

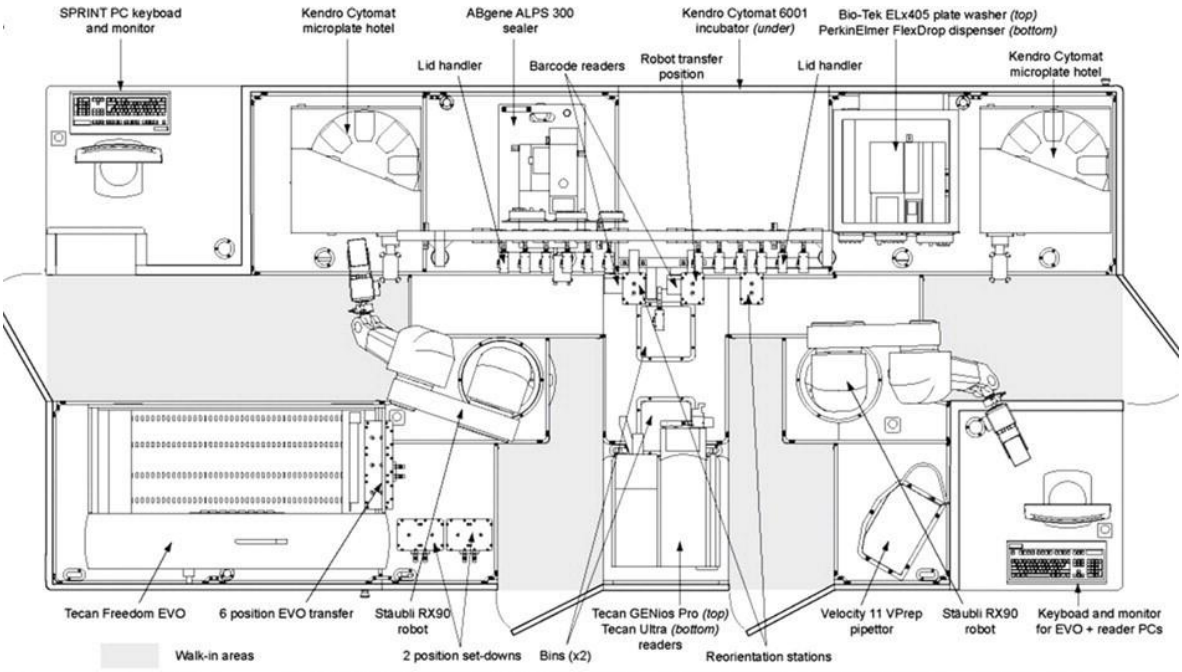


Figure 4: Stäubli RX90 robots for flexible screening processes [49]

Two robotic arms, Stäubli RX90 with a maximum payload of 11 kg and arm reach of 958 mm fitted with associated microtiter plate grippers are installed. The arms are primarily used for transportation of samples between the various instruments. The robotic arms work in two modes; combined mode when both robots work as one system and independent mode wherein two screens can be performed in parallel. Robot A has access to room temperature storage facility, 8-tip liquid handling station with shaker and plate manipulation and a plate sealer. The Robot B has access to a 96/384 liquid handling workstation, 96-1536 liquid dispenser and a 96/384 plate washer in addition to another storage facility. Both arms can access common instruments such as incubators, plate readers, etc. An underlying software called SPRINT creates dynamic schedules to be run in various combinations. Various third-party instrument software are linked to the SPRINT and triggered via secondary schedules. The system has been implemented at Terlings Park, Essex, UK for the parallel screening of four kinases against the same set of compounds. The kinases were tested to determine the duration of stability at room temperature.

2.1.3 RAMSAY-2 Dual-arm Robot

Processing of biochemical specimens often involve extensive manual work [50]–[52]. A group of scientists have developed a single arm system to robotize mass spectrometric analysis, involving initiation of biochemical reaction and subsequent monitoring to reduce the burden of sample processing and analysis on lab staff [53]. The system was further improved by the implementation of two synchronously operated robotic arms by Lynxmotion, Swanton, USA and is known as RAMSAY-2 [19]. Multiple microcontrollers were used for separate functions to facilitate simultaneous message processing. The robotic arms are controlled by open-source electronic modules based on the Arduino (Torino, Italy) concept. The two robotic arms, an RFID scanner, IR sensor, a message processing microcontroller and a touch sensor are each controlled by different printed circuit boards (PCB) which communicate within each other. The controlling algorithms are encoded in the C language using Arduinos free integrated development environment and are aimed at processing the signals from the various sensors and guiding movement of servo mechanisms in the arms. Arm 1 comprises of 6 servo motors while Arm 2 comprises of 5 motors and are programmed to transfer sample between processing stations. Both the arms carry out distinct operations making it possible to perform different operations simultaneously. The RAMSAY-2 robotic system was applied to demonstrate the screening of samples for enzymatic activity. The system performs sequential sample processing by recognizing the different samples, aliquoting solutions, incubating samples, delivering them to the mass spectrometer and initiating data acquisition (Figure 5). The sample vials placed in the drop off zone are recognized based on RFID tags. The tag stores information of the method to be carried out. The aliquots are added to the reagents automatically using microfluidic pumps. Arm 1 then transfers samples to a water bath for incubation and then places it on the transfer position. Arm 2 transfers the vials to the mass spectrometry interface where data acquisition is started after 1 minute and the spectra are acquired after another 60 seconds without human supervision. The touch sensor acts as a safety guard stopping the robot motion when any metal part is touched. The tasks of both arms are coordinated so that multiple samples can be processed simultaneously. The average processing rate for this application was recorded at 4 min/sample.

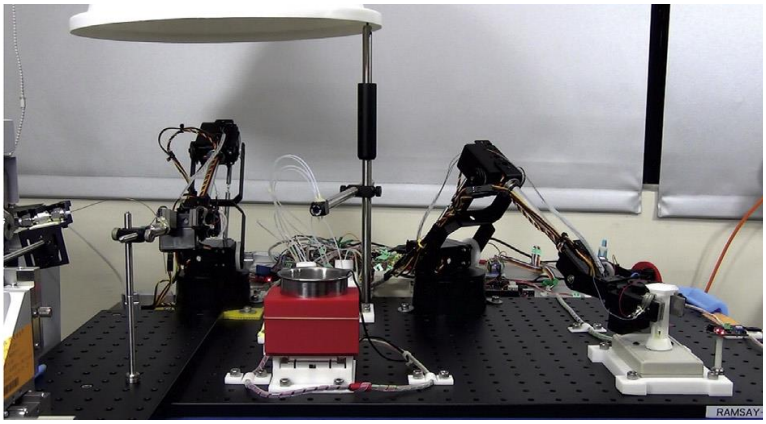


Figure 5: RAMSAY-2 robot for processing of biological samples [19]

2.1.4 ABB YuMi Cobot

Collaborative robots (Cobots) are robots which can interact physically with humans and work side by side in workplaces and offices. They have a lot of safety features and safety gears to make the interaction hazard free [54],[55]. Such robots can make a conversation, ask a question, answer a question, etc. They can be employed on a part of a production line for example and then hand over the part to a human for inspection [56],[57]. They are usually light, carrying smaller payloads of up to 35 kg [58] and can be easily reinstalled in a new space if necessary. They can also work on shop floor or production areas as industrial robots. They have the capacity to replace a human and take over all the manual tasks.

Cobots are a subset of intelligent assist devices and are the new robots of the future working alongside humans freely in a barrier free environment [60],[61]. All industrial robots until now have been designed to work independent of human interference and in their own robotic environments. Cobots however are designed to manipulate objects while also collaborating with humans [60], [61]. They reduce ergonomic concerns for assembly line workers while improving safety of workspace and quality of product. Workers can be relieved from repetitive manually stressful jobs by automating them [62]. Tedious tasks which require human cognitive skills, such as separating tangled work pieces like spring, segregating work parts can be done manually [63]. They also bring about manufacturing flexibility as a number of different part styles and motions can be included using the underlying software control [64]. Cobots can also assist workers in certain tasks such as lifting or moving big parts for assembly. The initial moving force or torque is provided by the robot and the worker only needs to guide the part to the destination. The worker is hence not overworked, and tasks can be finished quickly. The YuMi Cobot by ABB (ABB, Zürich, Switzerland) seen in Figure 6 is one of the most recent collaborative robots designed specifically to work alongside humans on the same task in a small parts assembly line.

Figure 6: ABB Yumi Cobot with dual arms [65]

It boasts of flexible hands and camera-based parts location system. The YuMi IRB14000 Robot has a reachability of 0.559 m and a payload of 0.5 kg per arm and a footprint of 0.19 m^2; thereby easily replacing a human in an already existing assembly line. The dual arms are extremely flexible with 7 DOF per arm (14 total for the robot) and weighs a mere 38 kg making it portable. It comes with an integrated controller, vacuum/servo fingers and can be mounted on a table surface thereby reducing robot footprint. It connects via ethernet for communication devices and other instruments. It has maximum velocity of 1.5 m/s and a positional accuracy of 0.02 mm. It has soft padded arms and a light weight construction to ensure co-worker safety. The robot uses lead-through programming, i.e.: the robot can be taught by manually moving the arm to the desired position and saving the coordinates for subsequent use [65]. The Yumi robot has been installed at the Texas Medical Center, Houston. Applications such as liquid transfers using manual pipettes, test tube handling and centrifuging

among others can be done by the robot. The installed system is currently in the phase of development and research. Current data suggests the robot is capable of performing processes 50 % faster than a trained human being.

ABB's mobile and autonomous laboratory concept robot 'YUMI' (Figure 7) is also developed for laboratory and hospital applications. The 14 DOF dual-arm robot can manipulate centrifuges handle test tubes and transport medical supplies. The robot can be implemented to perform tasks like transporting sheets and utilities between various hospital rooms, bringing medicines to patients at designated times etc. saving the nurses efforts.

Figure 7: ABB's YUMI laboratory and hospital cobot [65]

2.1.5 Andrew Dual-Arm Robots

With increase in high throughput and miniaturization of samples, liquid transfer is an important and challenging aspect of life science automation which contains many considerations for the practical application [66]. Liquid transfer techniques vary according to the density, viscosity of liquids as well as the volumes to be transferred. A number of methods such as gas driven and electrostatic force driven liquid transfer are used commonly in the industry. These automated systems can be used with a number of different assays [85],[86][69]. The other common method of liquid transfer is using the air displacement pipettes and electronically actuated pipettes [70]–[72].

Andrew Pipetting Robot using Conventional Pipettes

The Andrew pipetting robot (Figure 8a) was developed by Andrew Alliance (Geneva, Switzerland). It can automate laboratory liquid transfer tasks using conventional manual pipettes. The manual pipette of any volume can be mounted on the right arm of the robot while the left arm is used for manipulation. The left arm has a thumb like structure that actuates the plunger. The central vertical axis has a camera and rotation jig attachment. The camera reads the volume setting and the jig is used to rotate the pipette head to set the correct volume. Vision sensors set on the left arm detect bar coded placement tags to identify tip positions and labware positions. The robot is also assisted by force sensors to detect the proper loading of tips. The labware and tip holders are magnetically attached to each other to ensure proper placement on the workbench.

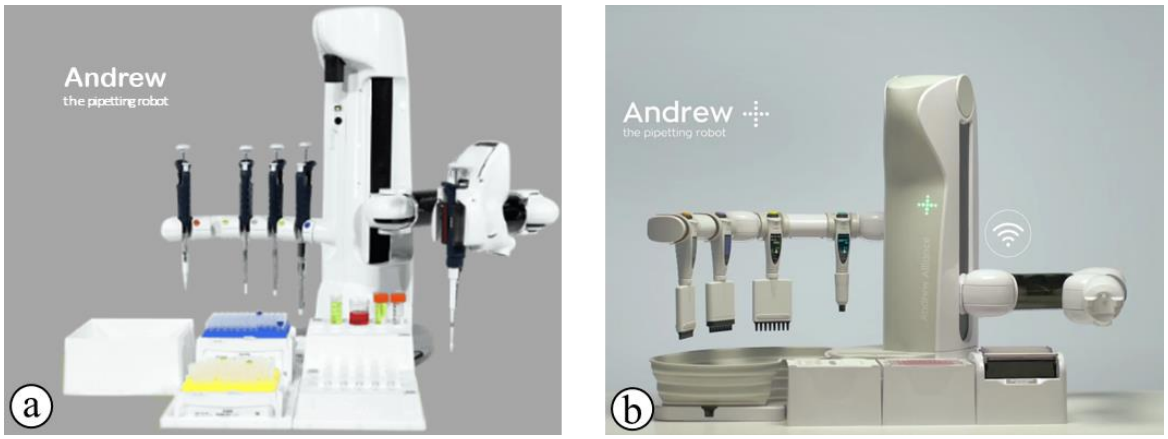


Figure 8: (a): Andrew pipetting robot with volume adjusting jig for manual pipettes, (b): Andrew+ robot for electronic pipettes [73]

The company recently also launched the Andrew+ robot (Figure 8b). It is a three-axis pipetting robot. Conventional electronic single channel as well as multi-channel pipettes can be used and it has a volumetric range of 0.2 µL to 10 mL. The process planning and scheduling can be done in depth using the company's OneLab software. The process details including labware types, sizes and positions, liquid volumes, pipette types, time duration for shaking and heating operations is set in the software. Depending upon the process, a deck layout is created by the software. The laboratory operator arranges labware according to the given layout. Microplates (MTP) and tips are placed in magnetic boxes, 'Dominos', to ensure correct alignment. The robot capabilities can be extended by adding up to 11 Dominos to perform shaking, heating and other operations. It follows vision guided positioning to validate the workbench layout, find the correct pipette and AI based object recognition for picking tips and identifying labware. The software communicates with the robot via a WIFI connection, cloud connection or bluetooth. The left arm picks up the required pipette from the right arm. The aspiration and dispensing volumes are adjusted automatically. It can also dispense liquids partially such that aspiration happens only once and liquid is dispensed in all wells of an MTP serially. The left arm of the robot can be adapted with a plate handling gripper, such that in addition to liquid handling, the robot can transfer MTP, perform tube/ MTP grabbing, chromatography column preparation and magnetic bead extraction. The robot can fit standard

lab benches and hoods and can also be placed inside a refrigerator for working with cold samples [73].

2.1.6 Other Dual-arm Robots Used in Laboratory Applications

Traditional cartesian robots have three mutually perpendicular principal axes and have 3 degrees of freedom (DOF). The motion is realized using sliding joints. Such robots are commonly used in industrial applications such as part assembly or pick and place applications. A SCARA robot is commonly two jointed robot arms compliant in the X, Y axes and rigid in the vertical Z axis. Such robots are used for high speed and high precision assembly and stacking applications. Traditional cartesian and SCARA robots are now being introduced in dual arm versions due to their multi-tasking capabilities. The dual arms are controlled by a single controlling unit and allow parallel processing reducing working time [74]–[78].

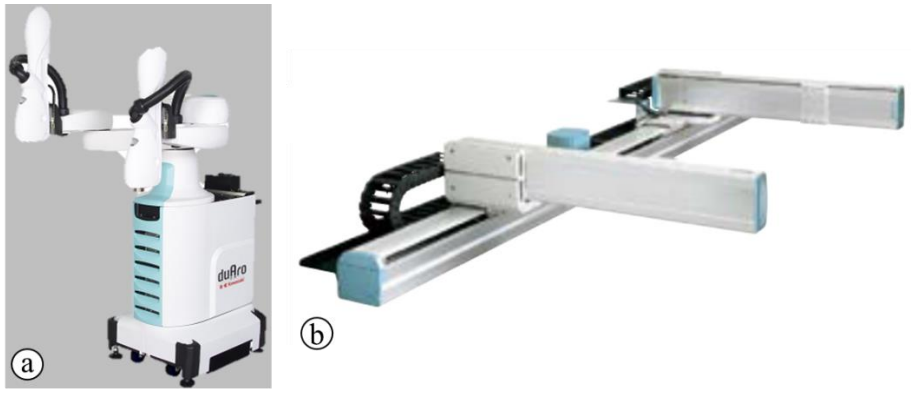


Figure 9: (a): Kawasaki duAro dual-arm SCARA robot [79]*, (b): Yamaha dual-arm cartesian robot* [80]

Tecan Cavro Omni Dual-arm Cartesian Robot

The Cavro Omni Robot (Tecan Trading AG, Switzerland) is a modular general purpose liquid handling system (Figure 10). The robot is highly configurable allowing any combination of XY, YZ, or XYZ axes with up to two arms per X axis. The system can be designed to have a single channel liquid displacement pipette, 2 single channel liquid displacement pipettes or a universal Z axis that allows the integration of multi-channel pipettes or custom gripper attachments. The Cavro robot can carry out precise liquid transfers in up to 1536 well plates. The positioning of the robot arm is done with the help of stepper motors with closed-loop control. The pipetting heads have an inbuilt capacitive level detector. The two arms can be moved independently or synchronously which helps improve throughput. The implementation of the Cavro Omni robotic system for automating the preparation of fungal samples for FTIR spectroscopy has been described in [81].

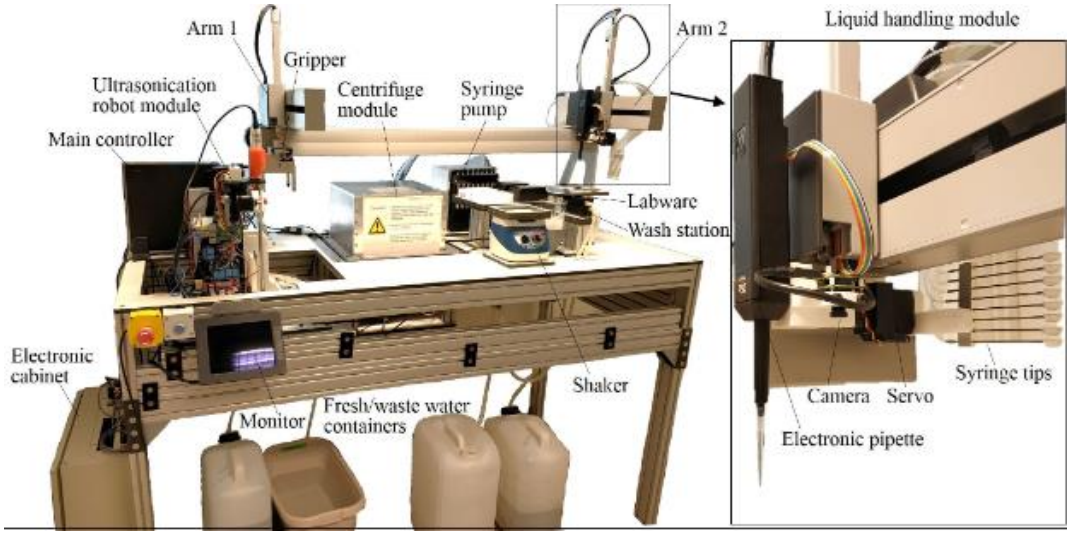


Figure 10: Tecan Cavro Omni dual-arm cartesian robot for automatio of preparation of fungal sample [81]

The developed robotic platform is an integration of three modular platforms: the ultrasonic module, centrifuge mode and the liquid handling robot. Machine vision enables full automation of the system without manual pre input information. The camera on the robot is used for detecting the number and position of well plates, differentiating between sample and blank wells. The left arm of the robot is connected to all the modules. The gripper attached to the left arm transfers 96 well MTP between the three modules. The right arm is fitted with an 8-channel syringe pump, electronic pipette and camera, a wash station and plate shaker. The sample spotting and washing require different volumes for which electronic pipettes and syringe pump are used respectively. To avoid collision, a servo motor is used to rotate the syringe tips to a horizontal position when not in use. All the hardware modules and components are connected via ROS (Robot Operating System, open-source robotics middleware suite). A main controller controls all the hardware modules. A dual-arm server node is developed to decode and encode the positions, speed and gripper operation. Actuation of the pipettes is done via an Arduino microcontroller. The accuracy of labware from identification with the camera was measured on the basis of images of 70 MTP, 60 IR plates, 270 tips and 80 blanks. The entire test was conducted twice with one MTP plate and two MTP plates which required a total of 942 minutes.

2.1.6.1 Dual-arm Cartesian Robot for In-meso Crystallization of Membrane Proteins

A dual-arm cartesian robot is implemented for the in-meso crystallization of membrane proteins [82]. The modular Xantus liquid handling robot (Xantus, New Castle, USA) was used for this purpose. The horizontal deck is loaded with two independent arms as seen in Figure 11. Arm 1 is used for liquid transfer. It is equipped with four small-volume liquid handling tips. Arm 2 was originally designed for plate transfer between different stations but was modified to handle viscous protein/lipid mesophase by replacing the griper with a positive displacement syringe. The syringe was connected to the robot controller via RS-232. New glass based 96-well plates were designed for this application. The crystallization trials were performed with 50 nL protein/lipid mesophase. A standard MTP can be set up in 13 minutes under standard conditions. The lipid mesophase was dispensed in wells of a 96 well plate and performance of the robot was measured with direct imaging.

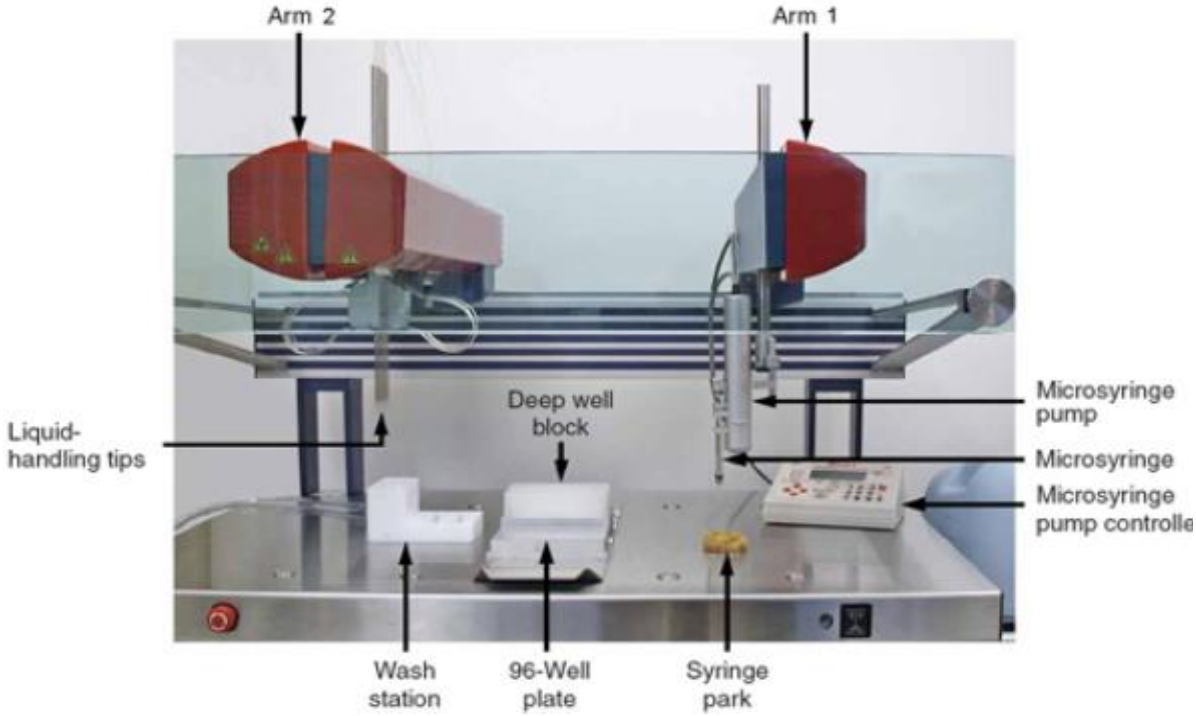


Figure 11: Xantus dual-arm cartesian robot for in-meso crystallization of membrane proteins [82]

2.2 Motion Control of Dual-arm Robots

Path planning of a dual arm robot is an important topic in the development of an automated system as the environment often consists of multiple obstacles that have to be avoided. The motions also need to be as direct and fast as possible while being dynamic in nature to accommodate the constant changes in robot surroundings. In the past years much research has been done to develop fast and intuitive solutions to the various challenges in path planning [83], [84]. The dual arms add to the complexity of such tasks due to the multiple degrees of freedom and intersecting reachability spheres which can cause collisions between the two arms themselves.

2.2.1 C-space Mapping Using the A*, RRT* Algorithms

A lot of research is being done currently in the sample based heuristic methods of path planning where a random sample of a larger grid is selected to determine the direction of motion [85] These methods provide an acceptable path within a reasonable amount of time and computational effort [86]–[92]. Sample based path planning methods refer to selection of algorithms used to find the most direct and shortest path from the start point to the end point on the basis of cost functions [93].

Y. Fei et. al. introduced a path planning algorithm based on the configuration space (C-space) [74]. According to this method, a collision free path can be automatically generated by specifying the start and the end positions and the orientation of the robot arm. The dual arm SCARA robot is employed to validate the method (Figure 12a). The C-space of a robot is the space of possible positions a robot may attain. An object with rigid coordinates and orientation is converted into a point in the C-space. The C-space is divided into two subspaces: obstacle space and free space. The C-space obstacles are defined as parts of the C-space where one arm can collide with the other arm.

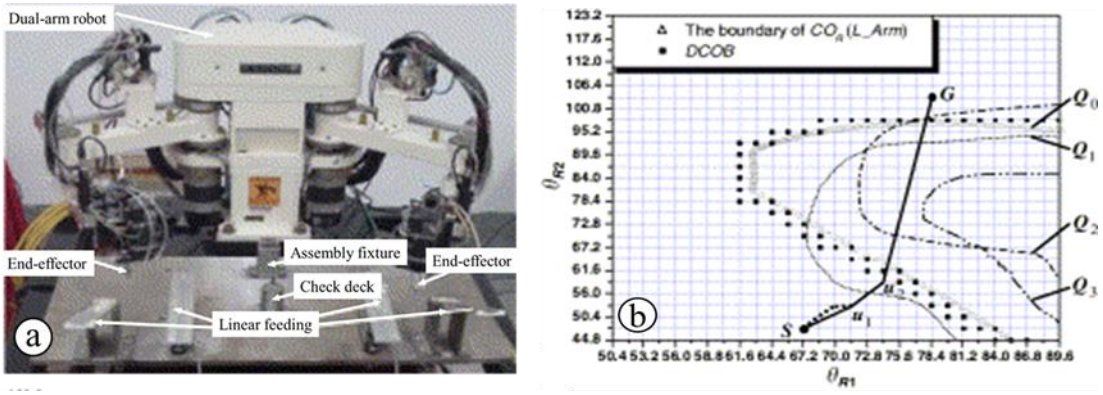


Figure 12: (a): SCARA dual-arm robot path planning using A heuristics, (b): C-space discretization. S is the start point, G is the target point, u1 and u2 are middle points.Q0,Q1,Q2,Q3 are the obstacle boundaries with robot arms at positions S, G,u1 and u2* [74].

The C-space is decomposed into a number of cells. The joint displacement of one joint when the other arm is static is selected as the decomposition criterion [94]. Based on these grids, obstacle boundaries are created corresponding to various arm positions. A start position and a target position is defined for the robot arm. The A* heuristic search method, which is a best-first search method, is employed to construct a path (Figure 12 b). The path is optimized using

the scan rule. A middle point of the path is found. The algorithm checks if the line connecting the present node and middle point intersects the obstacle space. If an intersection is not seen, the motion is continued. If an intersection is seen, the middle point is defined as the new start point and search is continued until the connecting line does not intersect with obstacles. This method gives a real-time collision free motion planning.

Another heuristic path planning approach is discussed by A. Perez et al [95]. Most sample-based path planning algorithms provide solutions that may not be optimal and cause jerky motions. This obstacle was tackled by introducing the RRT* (Rapidly-exploring Random Tree) method in [95]. The starting point is defined as a parent vertex. Random points (vertex) are generated in the plane of arm motion. The RRT* records the distance of each vertex. The closest vertex is selected and defined as a node. The neighboring vertices are checked, if a shorter distance is found, the node is replaced by the new node. More such nodes are created in a tree like structure. The distance between various nodes and vertices is continually checked. If a shorter distance is found, the previous node is immediately replaced altering the path of the robot. The RRT* generates straight paths and are useful in the presence of obstacles. This RRT* method was implemented on 12 DOF and 14 DOF robots to move the robot arms from under the table to the top of a table and pick up a cup (Figure 13).

Figure 13: Dual arm robot attempting to pick a cup from the table using the rapidly exploring random tree (RRT) algorithm. The image shows all intermediate positions of the arms while traversing from under the table to the target object [95].

2.2.2 Motion Planning Based on Gesture Recognition

With the increasing use of robots in the workplace, they are expected to replace human workers and replicate human motions and manual tasks. Various techniques of robot teaching by demonstration have been developed in the recent years. Human gesture imitation, however, is limited by the ability of control systems to replicate the force and stability of motion of the robot arm. Collision of arms with environment or self is another major concern. Various approaches such as following human motions using cartesian impedance control or simplified linear correlation of both arms have been developed to overcome these constraints [96], [97]. Dynamic environments pose additional challenges where the robot cannot be programmed for fixed tasks. Real time feedback from multiple sensors plays an important role in such scenarios as presented by Suh et al and Yang et al [98], [99].

A novel approach to achieving a coordinated task based control of a human directed 15 DOF Yaskawa dual-arm robot using tele-operated framework is presented in the works of D. Kruse et al [100]. The study setup includes peripheral sensors such as camera, Kinect module (Microsoft Corporation, Redmond, USA) and force/torque sensors in addition to the robot. The subsystems are integrated using in-house developed Robot Raconteur, which is an object-oriented distributed control and communication software system. The software is connected to all the modules and is responsible for interpreting messages and sending relevant signals. The system integration can be seen in Figure 14.

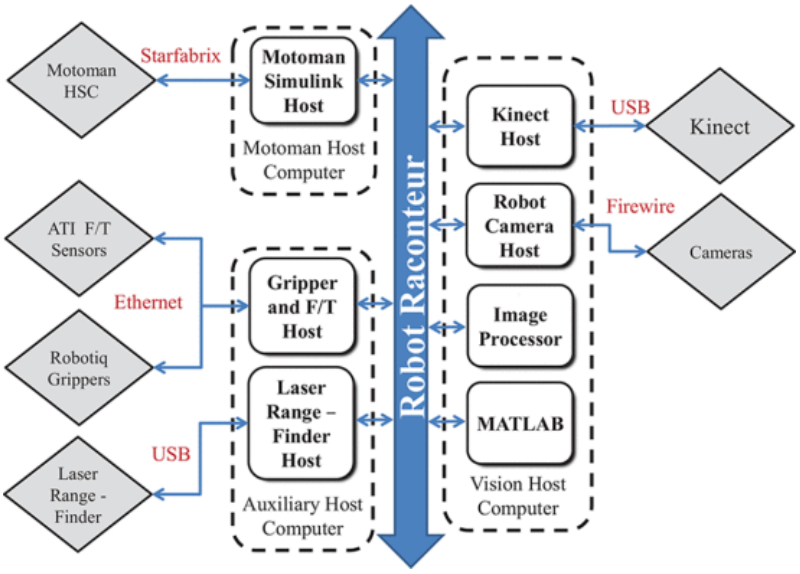


Figure 14: System architecture of the controlling system to direct a Yaskawa Motoman robot using gestures recognition [100]

The robot is expected to detect and grasp an arbitrarily placed object using visual servoing and transport the object to another location following operator gestures. The researchers implemented the damped least-squares algorithm to modify the joint motion. A Jacobian matrix mapping the 15 joint velocities to angular and linear velocities of both end effectors was generated. Virtual interference fields were defined and set as constraints to avoid collision of the two arms and a task space controller equation was formulated. The controller equation drives the end effectors to the desired location while avoiding undesired configurations.

The arm mounted camera was used for capturing image of the target object. 2D ALVAR tags (VTT Technical Research Centre of Finland, Finland) attached to the object were used for detecting the object to be gripped. The ALVAR is a library of virtual and augmented reality, used for creating marker-based tracking. The tags contain reference points that are mapped to the pixel positions in each image frame captured by the camera. The mapping gives information about the orientation of the object. Use of multiple tags reduces errors introduced during mapping. The task space motion controller mentioned above moves the robot arms to the target at a constant velocity until a defined contact force is created. The force is measured with the force torque sensor. The feedback from the sensor is adjusted in the control equation to achieve a stable grasp with stationary or moving object. After the object is gripped, the robot moves it to the destination. The Kinect SDK is used for this purpose. The Kinect can track 20 skeletal points of the human body and the motion is interpreted by the Kinect interface. A series of gestures helps the robot to perform tasks like, move, stop, grasp, release, go to home etc. The

overall system control and coordination is implemented in MATLAB and is not done in real-time. The process flow of such a system can be seen in Figure 15.

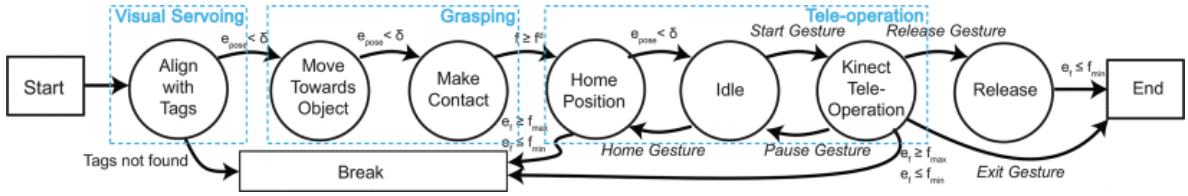


Figure 15: Standard process flow of detecting, grasping and gesture directed transportation of an object [100].

Makris et.al have described intuitive path programming developed in the Robot Operating System (ROS) [15],[85]. The system works on the principle of programming by demonstration. The system makes use of robot language including high layer libraries for motion control, gesture recognition, voice recognition and graphical user interfaces. The Kinect Xbox (Microsoft Corporation, Redmond, USA) is used for gesture recognition and oral command capturing. The process to be automated is fragmented into basic tasks such as approach, pick and place, etc. At the beginning of the process, the robot is made to record its position. The programmer then gestures for the motion, which is tracked by the camera and is converted to robot motion with the help of the various high-level libraries in ROS. The target position is again recorded (Figure 16). The robot can be performed in a smaller number of steps compared to traditional teach pendant teaching. The tool, frames, arm used etc. is all selected automatically by the libraries based on the type of task being programmed.

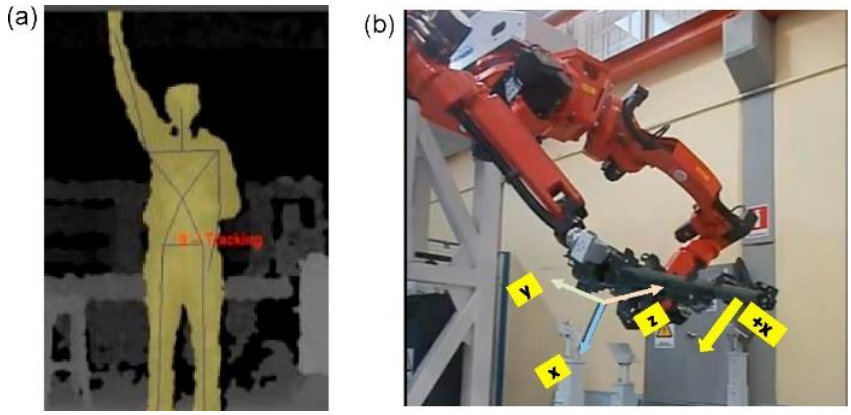


Figure 16: Dual-arm robot being taught using gesture recognition [12]

2.2.3 Constraint Based Task Programming

Basic automation applications such as transportation tasks or trajectory following in well-structured environments are straightforward and are performed regularly. However, unstructured and unknown 3D environments add complexity to the automation solutions and have not yet been researched extensively. Geometric intricacies give rise to unknown kinetic forces that may alter the motion of the robotic arm. Thus such robot motion planners make use of impedance control techniques by controlling the velocity, acceleration and force of the robot joints [102].

Sensors are commonly used for feedback and control in complex systems. There is no single systematic programming approach to deal with the sensors and the geometric uncertainty at the same time. Hence, J.Schutter et.al have developed a programming support for the implementation of complex sensor based robotic tasks in an unknown environment [103]. This background was utilized to implement a reactive force/velocity control of a sensorless ABB dual arm 7 DOF lightweight prototype robot [104]. The robot is position controlled. The axis controller receives joint position and velocity references from an external Linux real-time PC via a research interface.

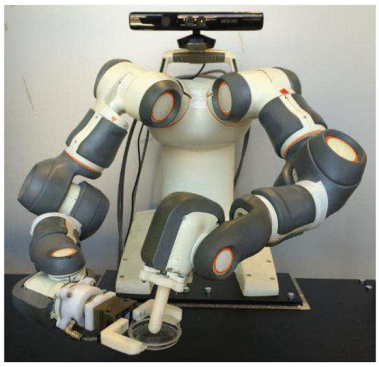


Figure 17: Dual-arm robot creating cell cultures with synchronous arm movements controlled via fuzzy logic [104]

The robot is not equipped with a force/ torque sensor. Hence, a model-based force observer was introduced. One arm (the leader) is considered to hold an object while the other arm is deemed the follower with a second object. The follower arm is required to follow a predefined path on a surface while maintaining constant contact and force on it. In the first part of the practical experiment, the left arm approaches the right arm holding the cell culture capsule. The second part is triggered by a contact force where the tool in contact with the capsule performs a circular motion while maintaining contact with the capsule. In order to accomplish the task, suitable bounds of maximum force and minimum force as well as the desired value of force (reference force value) of the interactive forces were specified. The model of the interactive forces was defined as equation (1),

$$f = K(z_0 - z), z \leq z_0 \quad (1)$$

Where f is the force exchanged, z and z_0 is the end effector coordinate and environment coordinate and K is the contact stiffness. The control model is derivatized and solved within the bounds of force constraints. The robustness of the system is affected by the robot position. In order to successfully complete the task, while still reacting and adapting to it, the control scheme is divided in two parts: the trajectory generation algorithm and the reactive controller. The reactive controller interprets the generated trajectory, combines it with sensory information and computes it to robot commands in the form of arm joint references. The reactive control algorithm implements a hierarchical quadratic programming (hQP) solver [105]. In case the robot is not able to follow the desired path, the error evaluation and the current position of the robot is sent to the trajectory generation algorithm as a starting point and a new path is computed.

2.2.4 Fuzzy Control of Dual-arm Robots

Fuzzy logic is a mathematical model with a non-binary solution. In the field of control theory, Jacobian matrices allow the approximate linearization of a non-linear system simplifying the solution to the controlled variable. Fuzzy logic systems converge rapidly in finite time and adapt to changes in the operating environment. This characteristic is used by Wu et al in their research of using dual-arm robots for sorting surgical instruments [106]. A dual-arm robot control scheme using the approximate Jacobian matrix and the adaptive fuzzy logic system has also been developed by C. Yang et al to control a robot with unknown dynamics and kinematics [107]. The research paper states that a dual-arm robot can be successfully controlled in the absence of robot dynamics and kinematics with the help of fuzzy control. The paper describes a dual arm robot commanded to grasp an object and follow a circular reference trajectory. The object's position and orientation vector are calculated for the joint variables of each arm and the Jacobian matrix is formulated in equation (2),

$$x = f_{kine_i}(q_i), \quad i = 1,2 \tag{2}$$

The position x of the object is defined as the function of q_i, which denotes the joint angle of the i^{th} arm.

The Jacobian is solved within the limits of the constraints for the kinematic parameters for each robotic arm. Similarly, the dynamic model of each robotic arm is derived from the lagrangian formulation of the above mathematical equation. Further, a multi-input multi-output fuzzy logic system is applied and a model is obtained that includes the approximation errors in the trajectory. To this model a partial permanent excitation condition is applied to ensure the convergence of estimated parameters and a solution in finite time. The resultant mathematical model is applied as the control strategy for the dual arm robot. The model was applied to a dual-arm robot to observe the performance of tracking trajectories and the errors therein. It was seen that even without certain robot kinematics and dynamics, the actual trajectories of the robot arm successfully followed the reference trajectories (Figure 18).

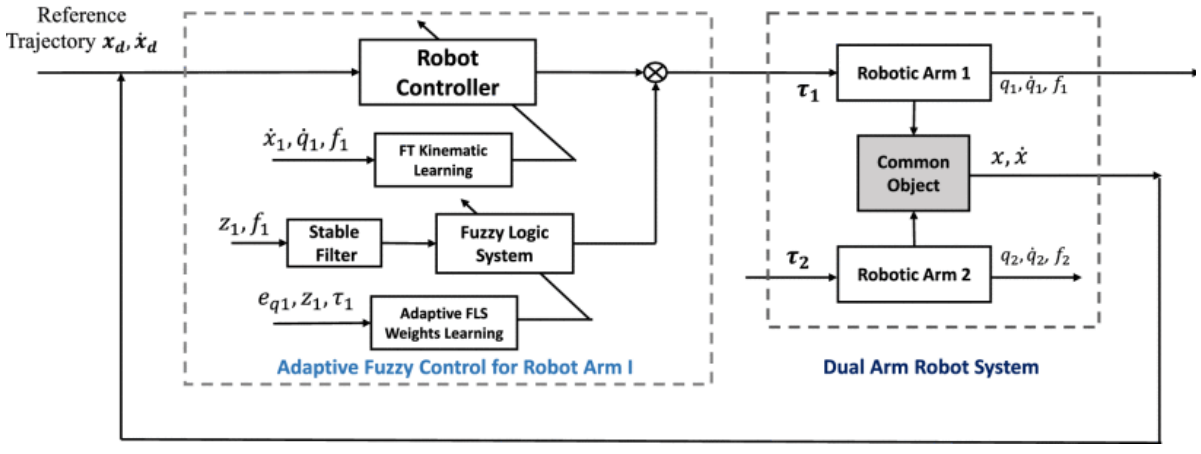


Figure 18: Fuzzy logic controller for the motion planning of a dual-arm robot in the absence of robot dynamic and kinematic information [107]

2.3 Summary

High accuracy, volume variability and task variation are characteristics of a laboratory process. Dual-arm robots with their multiple jointed arms can closely imitate human actions. Such robots can replace a human worker in high stress and toxic environments and can perform various tasks on routine lab equipment. Due to these advantages conventional SCARA and cartesian robots have also been introduced in the dual-arm format and installed for life science applications.

Tasks such as liquid transfer, shaking and homogenization, incubation, derivatization are important steps in a lab. Special instruments are available in the market to carry out such processes. Such equipment is highly specialized in a particular task requiring custom made labware. They also command a high monetary investment. Most of the literature related to current dual-arm application in laboratories has reference to a customized robot room with various automation friendly lab instruments for performing processes. They employ a number of sensory inputs to regulate and control the processes. However, it is observed, that these laboratory setups do not utilize commonly available labware such as manual pipettes, syringes or vials with screw lids. Instead, electronic pipettes and pop-lid vials are used. The robotic system described in this thesis addresses this shortfall by utilizing laboratory instruments and labware used for manual sample preparation processes reducing the need for special equipment. It performs continuous sample preparation of measurement samples for life science applications with minimal human intervention. Laboratory tests need to be accurate for optimum results. Thus, precise processes can be designed by laboratory personnel manually which can be replicated by the robot to ensure correctness. Using identical instruments reduces process variability caused by differences in instrument specifications or calibration.

The robot environment consisting of expensive instruments creates a complicated space for robot motion. Hence process planning and arm motion planning needs to be done meticulously. A number of sophisticated algorithms have been developed to move the robot arm around obstacles along the shortest path. Various heuristic algorithms are implemented for such applications in real time where the path can be changed dynamically mid motion if the position of the target object is changed. A number of systems are also seen to implement vision guided motion planning. However, these approaches would require complete change of the existing system and extensive testing which would be time-consuming. Hence, the presented robotic system uses a teach pendant to plan the robot motions. The robot motions are mapped on the basis of coordinate frames which are discussed in more detail in the following chapters.

3 System Overview

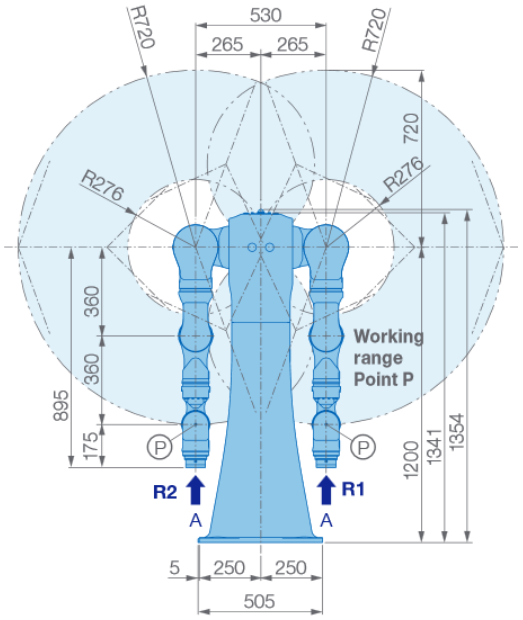


Figure 19: Yaskawa SDA10F reachability envelope [42]

The analytical laboratory in the Centre for Life Science Automation (Celisca) is currently equipped with a Yaskawa SDA10F dual-arm Robot (Yaskawa Electric Corporation, Kitakyushu, Japan). The robot boasts of 15 degrees of freedom (DOF) including 7 DOF per arm and 1 DOF to the body in the form of axial rotation around the Z-axis and a payload of 10 kg per arm. The robot joints are installed with a harmonic gear which gives a very high positional accuracy of ±0.1 mm and is preferred due to its zero-backlash error. The robot arms have a circular reachability of 720 mm (Figure 19). Dual-arm robots offer a higher advantage compared to single manipulators due to the human like dexterity [108],[109]. They are capable of handling multiple objects at a time and also pick up objects of small dimensions with precision [110]. The entire robotic system comprises of three parts namely: the robot hardware, labware and instruments, the interface program and the SAMI Ex (Beckman Coulter, Brea, California, USA)graphical interface software [111]

3.1 Robot Hardware

The dual-arm robot is used for classical sample preparation applications in a laboratory setting in order to replicate the tasks currently carried out manually by a technician. It is a part of a fully automated laboratory and hence requires to synchronize and collaborate with the other robots and instruments in the laboratory [112]. The dual-arm robot is placed in an enclosed space with all the instruments and labware placed around it. Figure 20 below shows the 3D rendering of the robot with all the surrounding instruments.

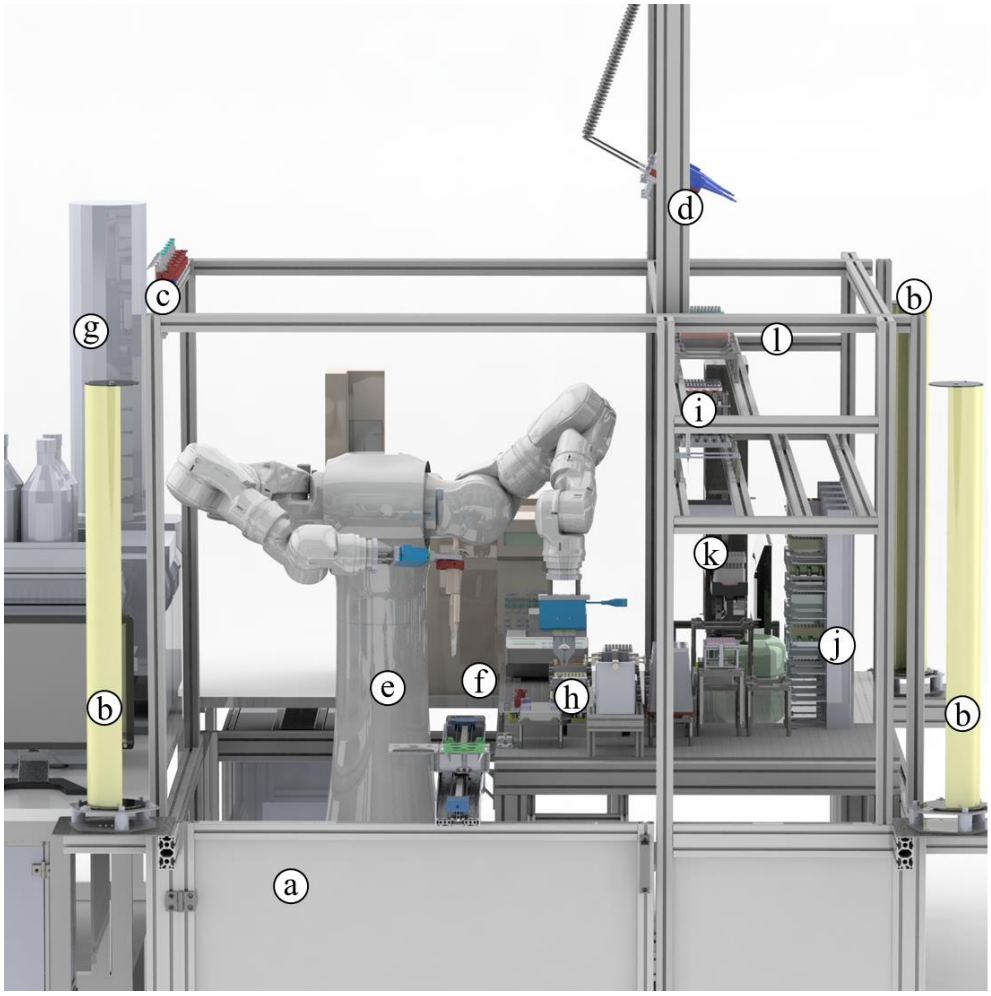
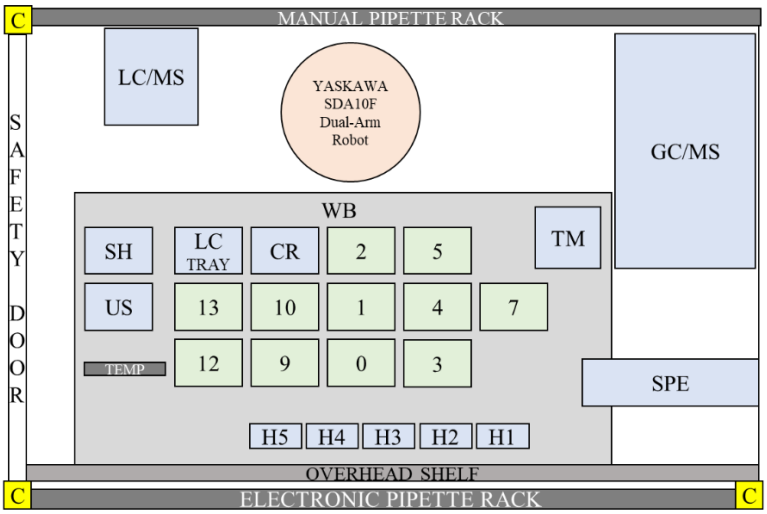


Figure 20: 3D rendering of the robot environment. (a) Safety door, (b) Infrared light curtain, (c) Manual pipette, (d) Electronic pipette, (e) Dual-arm robot, (f) Gas Chromatography Mass Spectrometer (GC/MS), (g) Liquid Chromatography Mass Spectrometer (LCMS), (h)Workbench with instruments, (i) Horizontal shelves, (j) Vertical storage, (k) Solid Phase Extraction (SPE)

In order to ensure safety of the operators, the enclosure is installed with a safety door. The robot stops mid motion if the security door is opened at any point during the working. As an added precaution, an infrared curtain is installed outside the enclosure, which acts similar to the safety door. The actual robot environment can be seen in Figure 21.

Figure 21: Yaskawa SDA10F dual-arm robot environment with instruments; (a) LC/MS, (b) GC/MS, (c) Manual pipettes, (d) Thermoshaker, (e) Workbench, (f) Cooling rack, (g) LCMS tray, (h) Shaker, (i) Ultrasonic bath, (j) Temporary pipette stand (k) Vertical storage, (l) Horizontal storage, (m) Electronic pipettes, (n) Dual-arm robot, (o) SPE

A number of racks are placed around the robot to store pipettes. Shelves are created to store plate type labware like MTP, tip boxes etc. In order to make optimum use of the vertical space, multilevel structures called hotels are also created to store and access labware. Positions of some of the instruments, like thermoshaker, ultrasonic bath etc., are fixed. Other labware can be moved around according to need. The exact placement of all instruments and workbench can be seen in Figure 22. The deck layout of the robot environment in RS-2 is the same as in RS-1. More hotel structures and a horizontal shelf has been added newly in RS-2.



LC/MS	Liquid chromatography mass spectrometry
GC/MS	Gas chromatography mass spectrometry
TM	Thermoshaker
CR	Cooling rack
LC Tray	LC sample tray
SH	Shaker
US	Ultrasonic bath
SPE	Solid phase extraction
H1-H5	Vertical storage (hotels)
0-13	Automated Labware Placer (ALP)
C	Infrared light curtain
TEMP	Temporary pipette stand
WB	Workbench

Figure 22: Top view of robot enclosure and relative positions of laboratory instruments in RS-2

3.2 Labware and Instruments

The greatest advantage of this dual-arm robot system is that the robot can manipulate all standard labware available in the laboratory. There is no need for investing in specialized labware, which leads to huge economic savings. Suitable labware adapters are designed and 3D-printed in-house to make the labware compatible with the automation environment and robot handling. The workbench is situated in front of the robot. A number of elevated structures called ALPs (automated labware positioners) are fixed on the workbench (Figure 23). Labware placed in the storage shelves is transferred to the fixed ALPs for the actual sample preparation process. The ALPs allow the labware to be placed in the exact position and orientation each time. The size of the ALPs and all other labware adapters perfectly matches the size of a standard 96 well MTP. Due to the standard sizes, the labware can be interchangeably placed on any ALP as required for the processes.

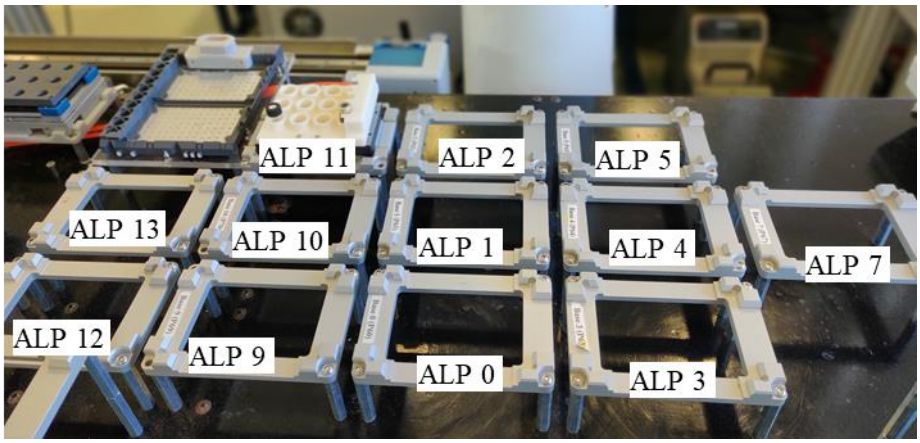


Figure 23: ALPs on the workbench

The workbench is utilized with 12 ALPs that are numbered serially (Figure 22). A cooling plate connected to a cooling pump is placed on the ALP 11. The plate is required when reagents have to be maintained below room temperature during the process. The cooling temperature is set manually. The ALPs may also be called 'Base'.

Manual pipettes (Eppendorf AG, Hamburg, Germany) are the most important instrument for liquid transfer and sample preparation process. The robot uses air displacement pipettes with nine different volumes that are hung on a rack behind the robot with the help of specially designed adaptors. Out of these, six pipettes are single channel and are used to pipette in or out of vials. The remaining three are multichannel pipettes that facilitate simultaneous liquid transfer to multiple wells in an MTP. The pipette volumes range from 10 µL to 10 mL (Figure 24). The dual-arm robot is not capable of reading the volumes on a manual pipette due to lack of visual input. A pre-programmed algorithm for setting the volumes of each pipette is complicated and time consuming. Hence, the required aspiration volumes of these air displacement pipettes have to be set manually before the start of each process. In order to overcome this limitation two electronic pipettes have been introduced in RS-2. Details about the electronic pipettes are given in later chapters.

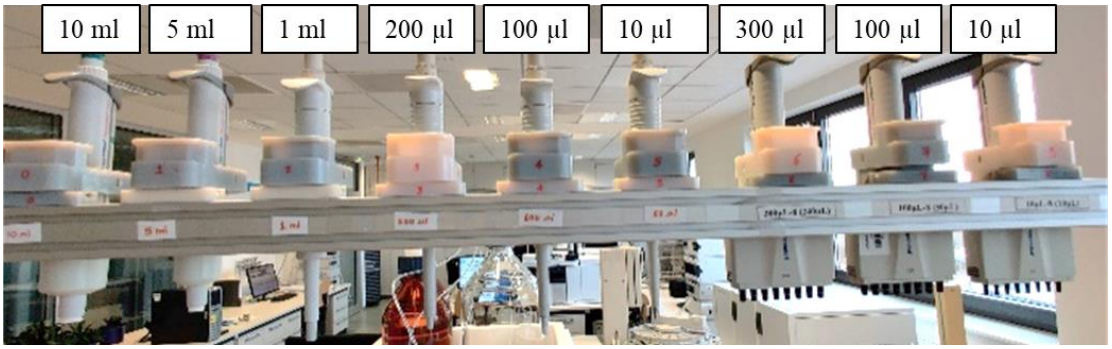


Figure 24: Manual pipettes hanging on a rack with the help of adapters for easy robot grasp.

An extra stand is fixed on the workbench where pipettes can be hung temporarily to be used again in an ongoing task (Figure 25). This stand is commonly required when pipetting a liquid alternately with two pipettes. This stand allows the pipette to be stored without discarding the tip. The stand in on the workbench, thus reducing the time required in picking and placing new pipettes.

Figure 25: Pipettes hung on a temporary pipette stand

Each pipette requires different sizes of tips as per volume. The tips are also sourced from Eppendorf AG, (Hamburg, Germany). Pipette tips up to 5 mL volume are contained in specially designed adapters (Figure 26). The dimensions of the adaptors match that of a standard MTP so that the tips can be placed on the ALPs during robot working. Tips with a volume of 5 mL and 10 mL have designated positions on the robot workbench as they require elevated ALPs due to the size of the tips. Tips up to 1 mL in volume are placed on two levels of horizontal shelves in front of the robot when not in use. For a liquid transfer task, the robot transfers the tip boxes from the shelf to base 7 (refer Figure 22) on the workbench before picking up the correct pipette from the rack.

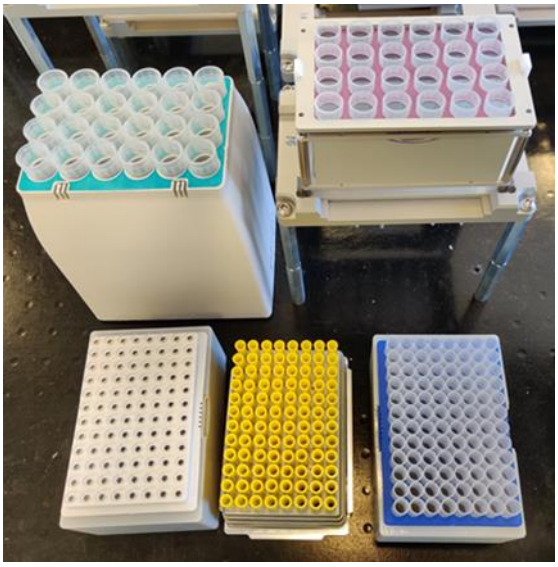


Figure 26: Various tip boxes in special adaptors

The two horizontal shelves are fitted with 15 ALPs in total. The shelves can be used for storing any plate type labware necessary for the sample preparation process. The labware capacity of each shelf is dictated by the robot reachability as the robot has a circular reachability envelope. Currently the robot is programmed to service 8 ALPs on the lower shelf and 7 ALPs on the upper shelf as marked in Figure 27b. Plate type labware such a MTPs, 2 mL vials, vial lids etc. are stored in the five vertical storage racks called a 'hotel'. Each rack has 8 usable slots, called 'room', providing 40 labware spaces in total. However, few of the rooms are inaccessible by the robot due to obstacles or incompatible arm geometry. Thus, a total of 31 rooms are currently programed to be used as depicted in Figure 27a. The rooms have a non-adjustable height limiting the type of labware that can be placed here.

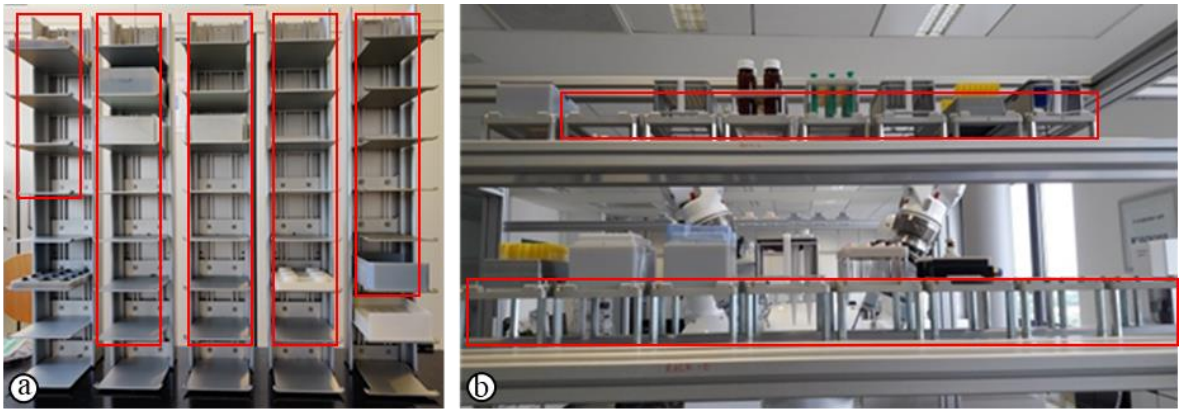


Figure 27: (a) Vertical Storage, (b)Horizontal labware storage

Five different types of vials (2 mL, 4 mL, 10 mL, 22 mL, and 40 mL) are currently in use to store the reagents. The vial holder with multiple vials is designed to fit the ALPs on the bench. The most common configuration is 4x3. Vials larger than 22 mL volume are arranged in a 3x2 configuration. Vials are required to be opened and closed by the robot multiple times during a process. Hence, matching lid racks are also designed to place the vials lids when not in use. The lid racks keep lids in a fixed position and suitable orientation so that the robot can easily grasp them (Figure 28).

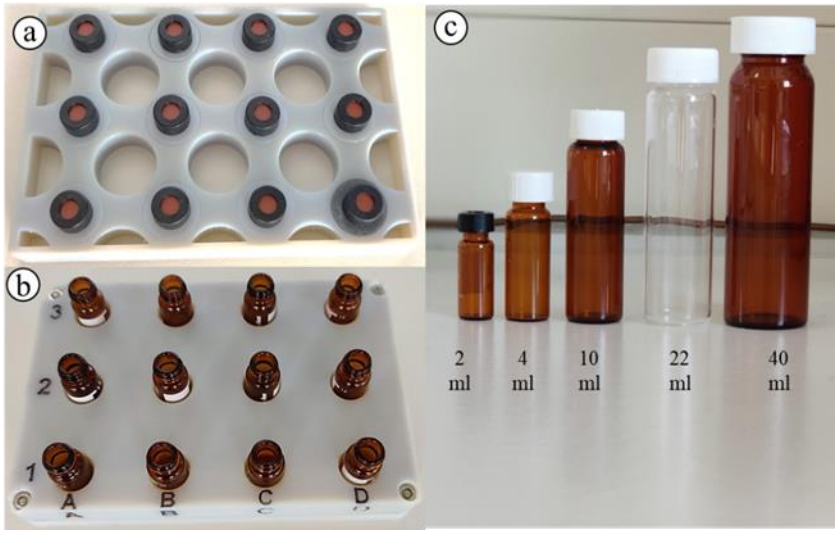


Figure 28:(a) & (b) 3D printed vial and lid holder, (c) Different sizes of vials

MTPs of various depth are used for sample preparation. Multi-channel pipettes are required for transferring reagents simultaneously to multiple wells of an MTP. These pipettes are incompatible with vials. Hence, reservoirs in the plate format are also available. The reservoirs are used to store reagents that have to be transferred to the MTPs with a multi-channel pipette (Figure 29).

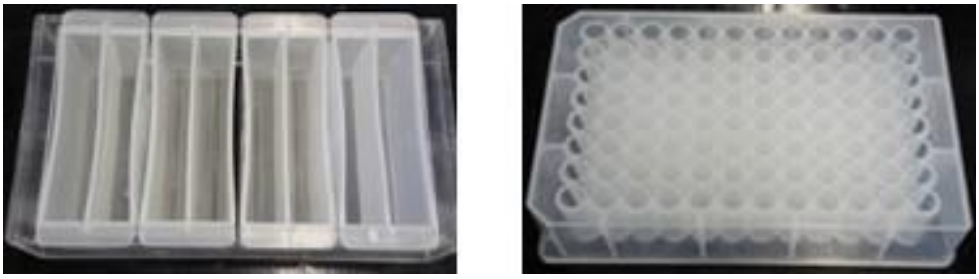


Figure 29: Eight tank reservoir and 96 well microtiter plate for sample preparation

The dual-arm robot is capable of using complex instruments such as syringes, glass pipettes and glass syringes. Special containers are designed to hold these labware as seen in Figure 30. The containers are designed to the size of a standard MTP and are meant to be placed on the ALPs during sample processing. Such labware is also stored on the shelves or in the hotels when not in use and transported to the workbench as required. The glass pipettes have a designated rack on the workbench.

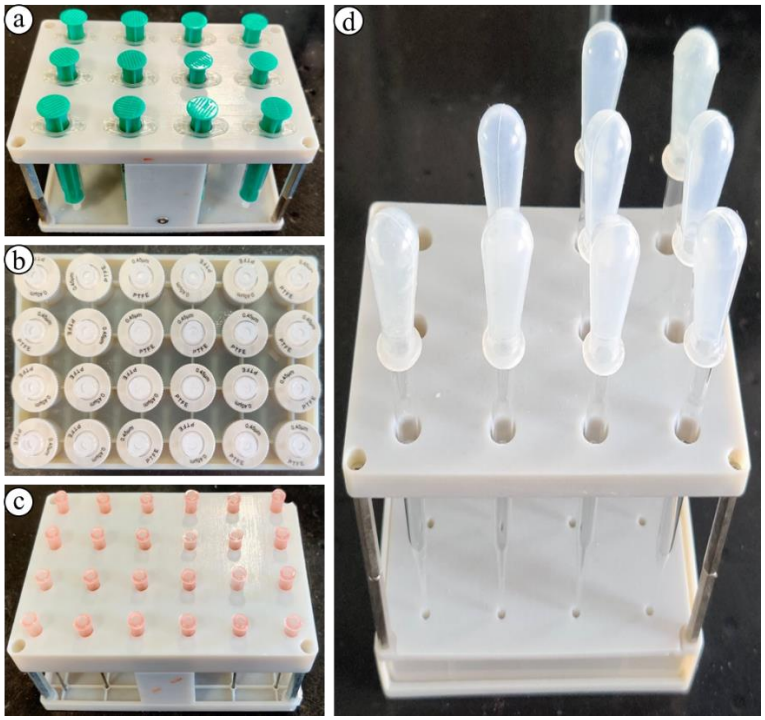


Figure 30: (a): Syringes, (b): Filters, (c): Cannulas, (d): Glass pipettes

The prepared samples have to be shaken to promote mixing and chemical reactions. Thermoshaker (Eppendorf, Hamburg, AG), shaker (Thermofischer Scientific, Massachusetts, USA) and ultrasonic bath (Bandelin electronic GmbH, Berlin, Germany) are used for this purpose (Figure 31). A cooling plate is also available for special requirements.

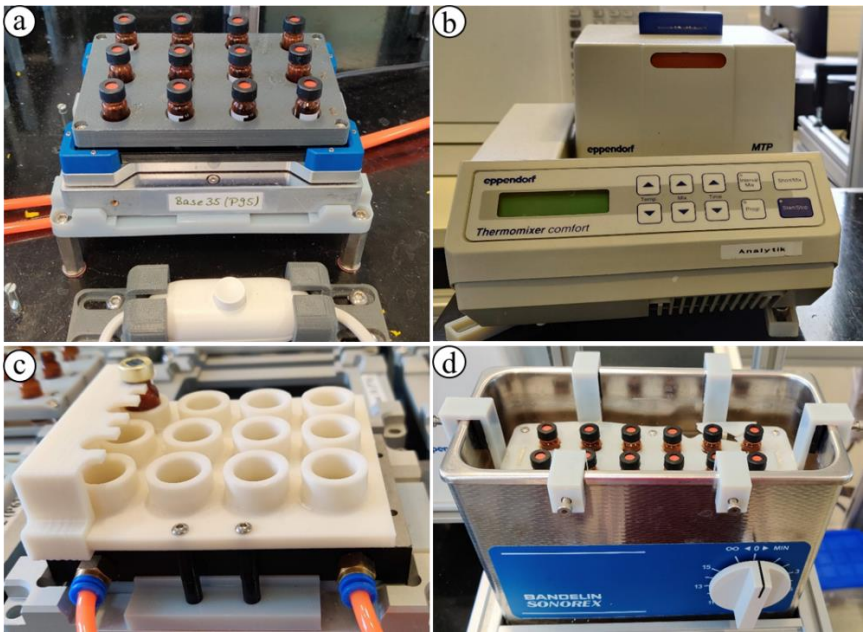


Figure 31: (a): Shaker, (b): Thermoshaker, (c): Cooling rack, (d): Ultrasonic bath

3.3 SAMI Ex User Interface

The SAMI Ex is a process designing and scheduling software by Beckman Coulter, Brea, California, USA. The user can design sample preparation and analysis processes with the software. In order to start the analytical measurements automatically, the robot controller, the GC-MS and LC-MS analytical systems are integrated with the SAMI Ex via a local connection (Figure 32).

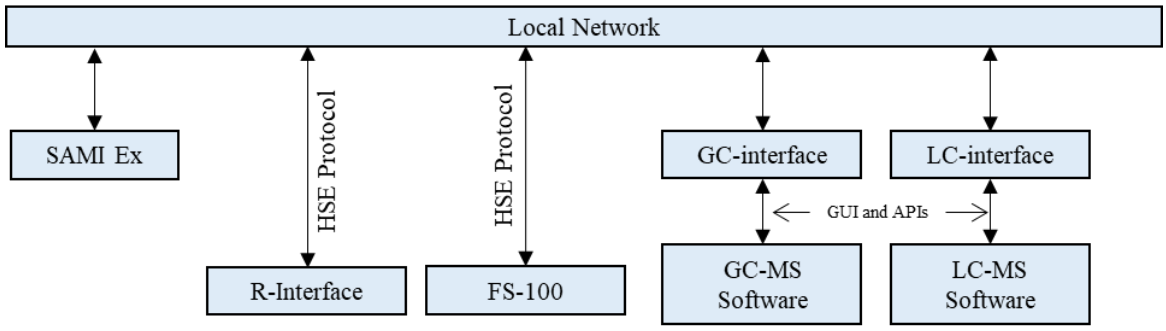


Figure 32: System Configuration

The LC-MS and GC-MS are controlled by the LC-MS data acquisition and GC-MS data acquisition software respectively that run on two different personal computers (PCs). The robot is controlled via the FS100 controller. In order to integrate the instruments and the robot with SAMI Ex, three independent interfaces have been built [111].

1. The LC-interface bridges the LC-MS data acquisition software with SAMI Ex.
2. The GC-interface bridges the GC-MS data acquisition software with SAMI Ex.
3. The robot interface (R-Interface) connects the robot via the robot controller to SAMI Ex.

The SAMI Ex user interface has an underlying SILAS kernel that is responsible for scheduling processes created in SAMI Ex and communicating with third party interfaces. The three interfaces named above directly communicate with the SILAS. Figure 33 depicts the architectural flow of the control system.

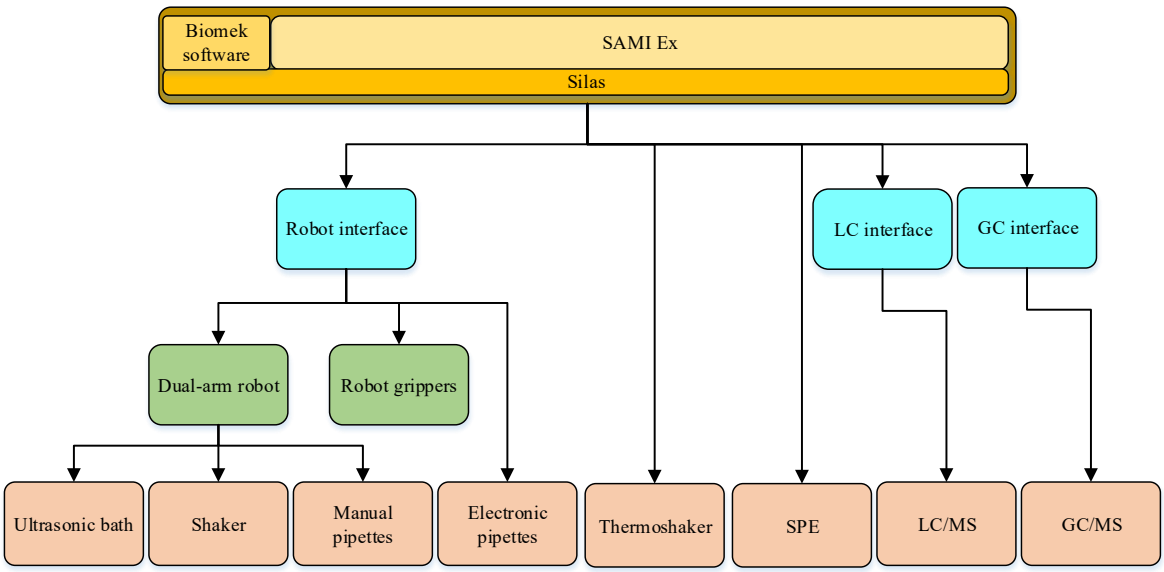


Figure 33: Architecture of the process automation system

The robot end effectors and the electronic pipettes take commands directly from the R-Interface. The thermoshaker and the SPE receive commands directly from the SAMI interface via a USB connection. They however do not need an interface and can communicate directly with the SILAS. The shaker and ultrasonic bath have on/off buttons that are manipulated by the robot.

The SAMI Ex graphical interface is designed in the form of blocks depicting the different labware and instruments. The Biomek software, from Beckman Coulter, Brea, California, USA, is a background software supporting method generation in SAMI Ex.

The communication between SAMI Ex and the interfaces is carried out using socket connections, which are two-way communication links between two programs running on a network. Two threads are used to handle SAMI Ex commands: the scanning thread and the processing thread. The scanning thread is responsible for getting commands from SAMI Ex. The first command passed by the SAMI Ex starts the processing thread. The processing thread checks if the command is valid. If yes, processing is started. While the first command is being processed, the thread cannot accept new commands. In the meanwhile, the scanning thread continues to search for new incoming commands. When the next command arrives, the scanning thread checks if processing thread is busy, if so, the scanning thread is set to wait. If not, the new command is sent for processing. Thus, only one command is processed at a time. Nine commands have been identified as valid in order to start the processing by the processing thread.

Table 1: Commands processed by the GC-MS interface, LC-MS interface and the R-Interface

Command Name	Command valid for			Task
	LC-MS	GC-MS	R-Interface	
Version	x	x	x	Get version of LC-interface, GC-interface and R-Interface
List Methods	x	x		Get list of GC-methods and LC-methods
List Projects	x	x	x	Get list of projects
Status	x	x	x	Get status of the instrument: idle, busy, error
MethodRun	x	x	x	Load worklist, GC-MS, GC-MS method and start a measurement. This command is used for pipetting and syringe tasks in the R-Interface.
Move			x	For transporting labware between shelf, hotel, LC tray and workbench. Open or close thermoshaker lid.
Lid			x	Put lids to vials or cover MTPs
Delid			x	Remove lids from vials or MTPs
Abort	x	x	x	Abort measurement or abort robotic process
Stop	x			Stop LC-Ms measurement
Continue	x		x	Resume LC-MS measurement. Resume robot movement.

With regards to the R-Interface, the processing thread extracts information from the command and organizes it in an XML format. The XML data is used to generate a call file containing the various robot motions. The call file is then downloaded to and executed by the FS100 controller to perform robotic tasks. The 'MethodRun' command contains information such as task type (pipette task, liquid extraction task). It also contains information regarding the labware to be used and the labware positions. Similarly, the 'Move', 'Lid', 'Delid' commands contain information to the labware source and destination positions and labware types.

A SAMI method is a partial or complete sequence of tasks required for the sample preparation process, designed in the SAMI Ex software. The user defined labware icons (Figure 34) created in the Biomek software are imported in the SAMI Ex in the form of graphical icons.

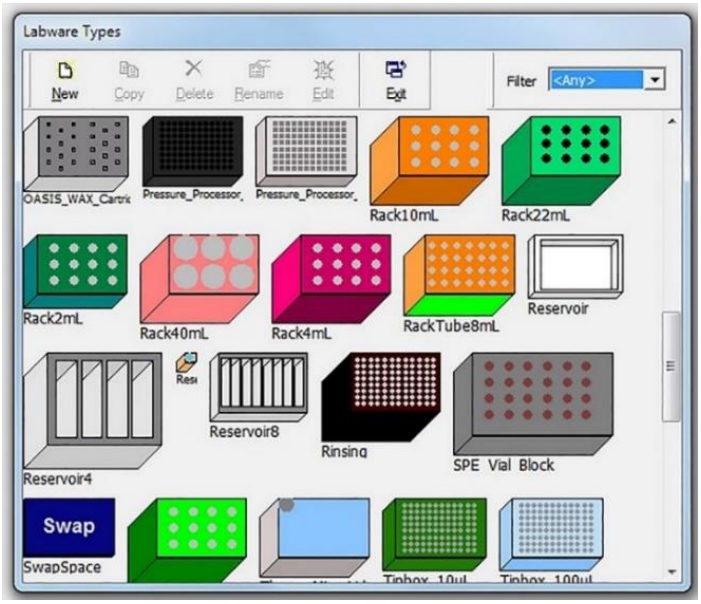


Figure 34: New labware definition for importing into SAMI Ex methods

Table 2: List of labware descriptions as seen in the Biomek software

Labware defined					
Labware Name	**Description**	**Labware Name**	**Description**	**Labware Name**	**Description**
MTP ***96***	*96 well* MTP	***LidRack_2mL***	*2 mL vial lid rack*	***Rinsing***	*SPE rinsing block*
Reservoir4	*4 cell tank*	***LidRack_10mL***	*10 mL vial lid rack*	***SPE Vial Block***	*2 mL vial adaptor (8 x 6)*
Reservoir8	*8 cell tank*	***Tipbox_10uL***	*10 µL tips*	***Cannula***	*Cannula*
Rack2mL	*2 mL vial rack*	***Tipbox_100uL***	*100 µL tips*	***Filter***	*Filter*
Rack4mL	*4 mL vial rack*	***Tipbox_300uL***	*300 µL tips*	***Syringe***	*Syringe*
Rack10mL	*10 mL vial rack*	***Tipbox_1,000uL***	*1 mL tips*	***GP***	*Glass pipette*
Rack22mL	*22 mL vial rack*	***Tipbox_5mL***	*5 mL tips*	***US***	*Ultrasonic bath*
Rack40mL	*40 mL vial rack*	***Tipbox_10mL***	*10 mL tips*	***TSIM***	*Cooling plate*
MTPLid	MTP *lid*	***Cartridge***	*SPE cartridge*	***GC***	*GC*

The programmer can select the required labware from a list of the various labware configurations available, which can be seen on the screen as a picture. The configuration icons can be created by the user as per requirement. All labware required for the process can be selected in this manner.

The labware and instruments are connected with unidirectional arrows showing the direction of flow of process. A number of properties such as the home position, which is the default position of the labware, can be set for each of the icon. Additionally, other properties such as destination positions or the number of vials to be picked up from a rack etc. can be specified. Additional details regarding the labware to be used, tip size to be used etc. are also specified by the programmer. Additional information pertaining to a specific task, such as pipette specifications, or button pressing is specified in a designated block (Figure 35). Icons are also available for the various lab instruments used. The duration of actuation of these instruments can be specified in the SAMI Ex software.

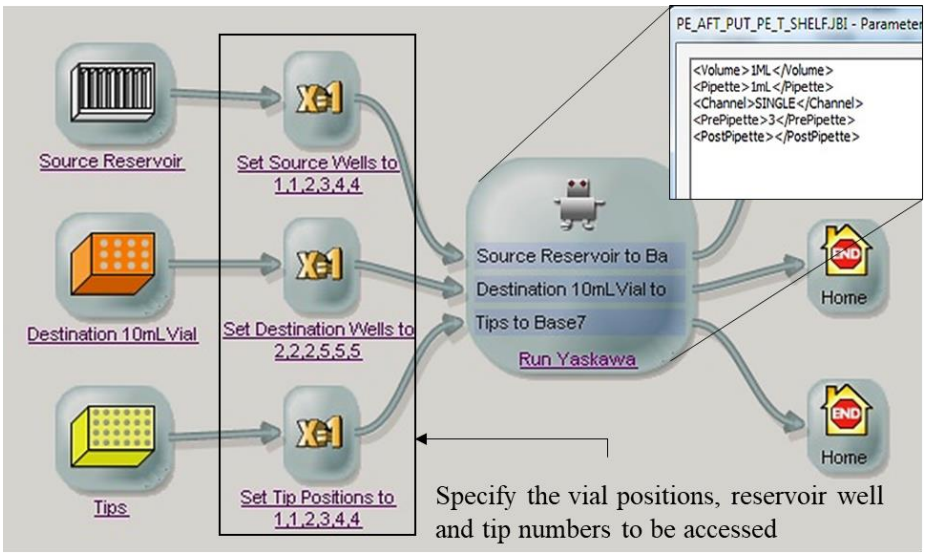


Figure 35: Depicts the specifications of a pipetting task using manual pipettes. The 'Volume' and 'Pipette' fields are used to identify the pipette type, 'Channel' field differentiates between the single and multi-channel pipettes, 'PrePipette' field is set if pre-wetting of the pipetting tip is required.

The end of a process is depicted by an “end” icon to indicate there are no further commands to be carried out. The programmer can see and manually check the flow of all the labware and tasks to be carried out during the process. The different tasks in the sample preparation process are programmed as individual SAMI methods in the RS-1. Thus, if a process contains six tasks: vial opening, pipetting 1, pipetting 2, vial closing, thermoshaker, sample feeding to LC/MS; six method files will be generated for one process.

The SAMI Ex scheduler runs the process offline to check for possible errors. The scheduler output is shown in the form of a bar diagram as in Figure 36. Each bar corresponds to each labware. The presence of each bar at particular time on the graph tells us the position of the labware at that time print.

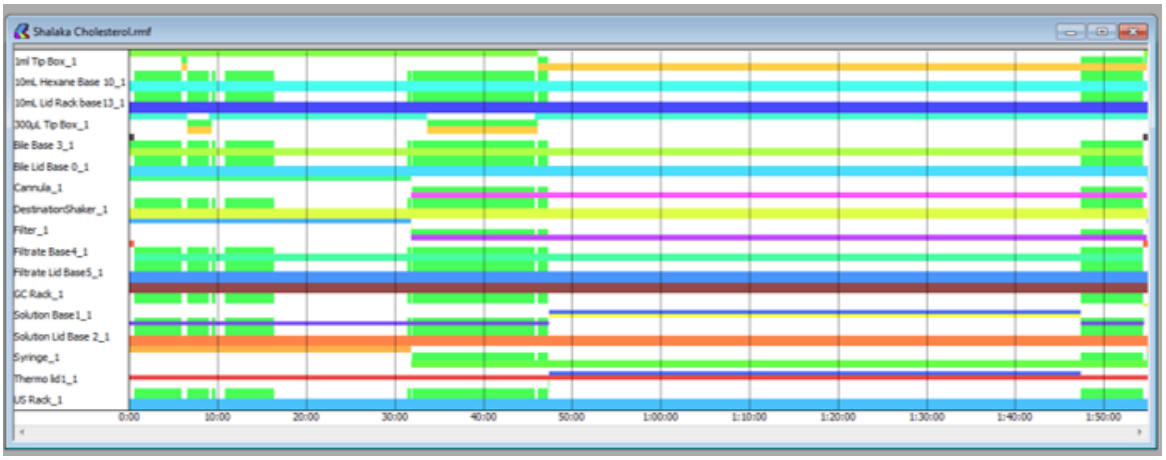


Figure 36:SAMI Ex Method Scheduler for process verification

In case of ambiguities in the process or collisions between labware, the software raises an error alarm. Errors can happen if the programmer has set two labware in one position at the same time, or if a variable value is set outside of the permitted range. The scheduling property is very useful to avoid errors during the robot run and possible breakage of labware or wastage of reagents. The finished SAMI Ex method is executed with the help of the SAMI Runtime software which compiles the method and sends execution commands to the robot controller via the R-Interface in the form of XML commands. An overview of the entire process can be seen in the runtime window. The screen shows the position of each labware in real time and highlights the current ongoing task (Figure 37). The different labware and labware properties are listed in the runtime window in order to verify the process.

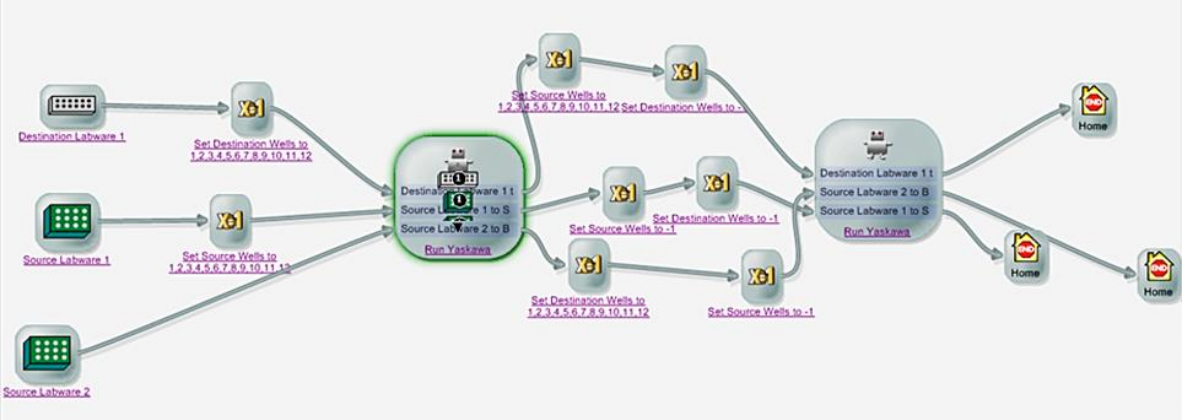


Figure 37: Process run of an ultrasonic shaker task in SAMI Runtime software. The process status can be observed remotely in real-time by tracking the motion of labware icons.

The individual method files are difficult to edit as multiple files have to be referenced to guarantee continuation of the process. Creating individual method files for different tasks is time consuming and can potentially introduce many manual errors, such as a labware placed in different positions in consecutive tasks, etc., which affect the flow of tasks. Hence, the SAMI method creation has been changed in the RS-2. The changes have been outlined in a later section.

3.4 Robot Interface (R-Interface)

The robot interface (R-Interface) is a program developed in-house [111] which allows the SAMI Ex software to communicate with the robot. It is written in the C# language of the .Net framework. The R-Interface comprises of multiple modules such as the communication module, job database, message processor and various task modules such as shaker module, thermoshaker module, pipette module, SPE module etc. (Figure 38).

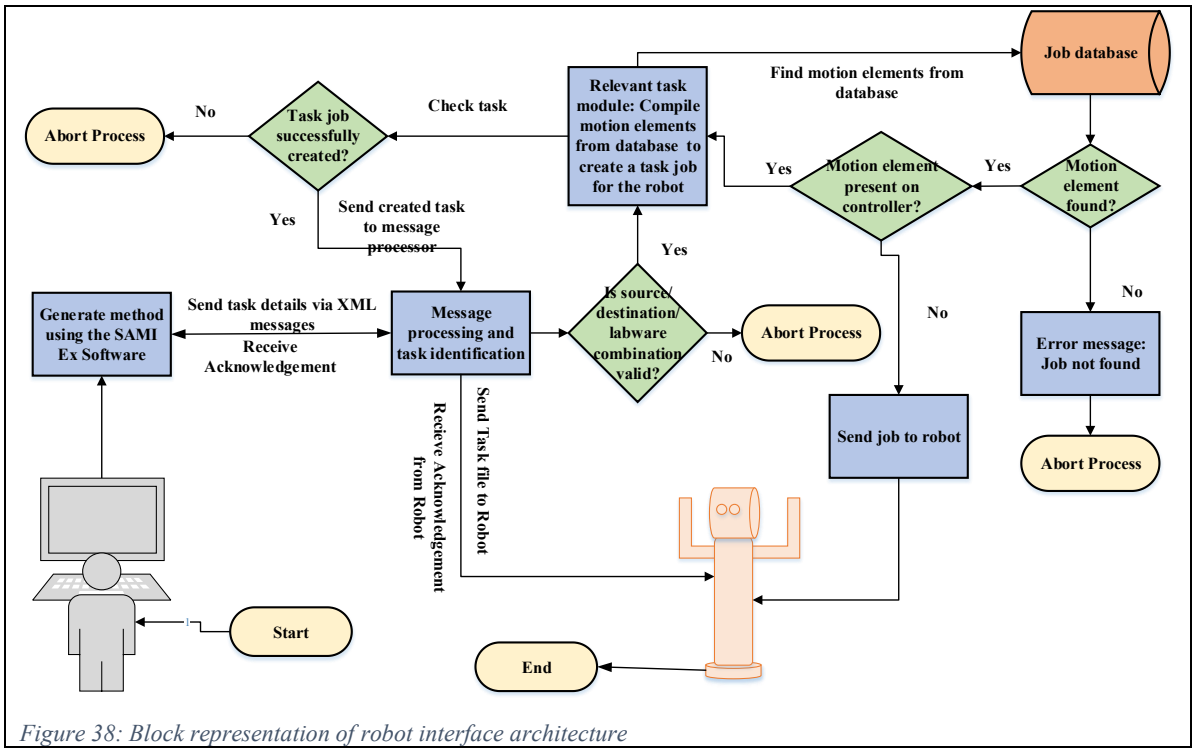


Figure 38: Block representation of robot interface architecture

The communication module primarily takes care of the sending and receiving message packets between the robot controller, R-interface and the SAMI Ex user interface. This module does not decode any of the messages but passes them in the form of an XML file via an Ethernet connection. The messages sent contain information for turning the gripper or instruments on or off, or the information about a robotic motion to be carried out. An example of the XML message received from the SAMI Ex by the R-Interface is seen in Figure 39.

```
<?xml version="1.0" ?>
- <Yaskawa-Commands ID="47" Type="Command" Client="Yaskawa">
- <MethodRun CommandName="MethodRun" ProjectPath="Ultrasonic" MethodName="Ultrasonic
Method" DataPath="Mond.links\oben" RunState="1">
- <Parameters>
  <Command>Move</Command>
  <Time />
  <SourceWells Labware="DestinationShaker_1">1,2,3,4,5,6,7,8,9,10,11,12</SourceWells>
  <DestinationWells Labware="US Rack_1">1,2,3,4,5,6,7,8,9,10,11,12</DestinationWells>
  </Parameters>
- <PosData Name="Shaker">
  <LWData Name="DestinationShaker_1" BC="" SamiClass="Rack2mL" />
  </PosData>
- <PosData Name="US">
  <LWData Name="US Rack_1" BC="" SamiClass="Ultrasonic_Rack2mL" />
  </PosData>
  <WellData />
  </MethodRun>
  </Yaskawa-Commands>
```

Figure 39: XML communication between the SAMI Ex and R-Interface for a vial transfer task between the ultrasonic shaker and mechanical shaker

The message processor module encodes information from the commands received from SAMI Ex and generates replies and execution commands for the robot controller. Once the message is deciphered and the robot task is recognized the individual task modules are triggered. For example: the code in Figure 39 contains an XML file for transfer of vials from the shaker to the ultrasonic bath. The code contains information to the source and destinations positions of the labware as well as which vial numbers should be transferred. The message processor will read the 'MethodName' which is 'Ultrasonic Method' and directs it to the 'Vial task module'. The vial module manages transfer of vials between various positions.

The task modules contain specific details of the task such as the allowed labware, position of the labware, allowable labware configurations, values of various variables used etc. Each sample preparation process is created by combing multiple tasks, like labware transportation, liquid handling, sample mixing, etc. Every task is divided into a number of sub-tasks, called motion elements or robot jobs, which are robot arm motions. In the above example for an ultrasonic module, the vial task module is activated. Here, the vial number, labware positions etc. will be validated. Variable indices will be assigned to the labware number, type, position etc. The variable indices make the command readable to the robot controller. A sequence of motion elements to carry out the task is specified by the programmer. This sequence is unique to each task module. Accordingly, motion elements are recalled from the database. The motion combination module sets all the recalled motion elements and the variable indices in a specific order and saves it to a file, called a 'task file'. The motion elements (jobs) and task files are .JBI files, which is the only file format that can be read by the controller. The communication module then transfers this file to the robot where it is run and the motions are carried out. The task modules are so designed as to be independent of each other. Change in the motion elements or structure of one task does not affect any other modules. New independent tasks can thus be added as required without disturbing the system. After the tasks are finished, the communication module sends back a message indicating the completion of the task and the control is transferred to the R-Interface again where the entire process is repeated for the next task.

4 Aims of Dissertation

The Yaskawa SDA10 dual-arm robot has been programmed to perform a number of tasks, for example, labware transport, pipetting, liquid extraction, etc. The robot can handle labware such as vials, pipettes, glass syringes, and thermoshakers in the same way that a human does. Instruments such as the GC and LC that are used to analyze the samples can also be accessed by the robot. These motion elements are programmed using the teach pendant of the robot and define the actual robot motions. However, the automated system that was already in place and running (RS-1) was seen to have certain limitations such as being time consuming and unable to process multiple consignments of samples in one process. Hence, the system has been optimized in a number of areas such as quantity of samples processed, creation of motion elements, etc. and an optimized system RS-2 has been developed. In the RS-1, nearly 262 motion elements have been created for the robot to perform various tasks. Individual motion elements are created for individual labware types. The FS100 controller memory limits the number of motion elements saved in the controller at a given time. Hence, prior to starting any sample preparation process with the robot, it needs to be checked if all the required motion elements are present on the controller. Else, they have to be loaded manually. A fixed sequence of motion elements is hardcoded in the R-interface in order to carry out a task. The sequence starts from the standby position (Figure 64) of the robot and ends at the standby position. The robot arms thus travel to this position very frequently during the process making it aesthetically unappealing and time consuming. The general overview of the system flow is depicted in Figure 40.

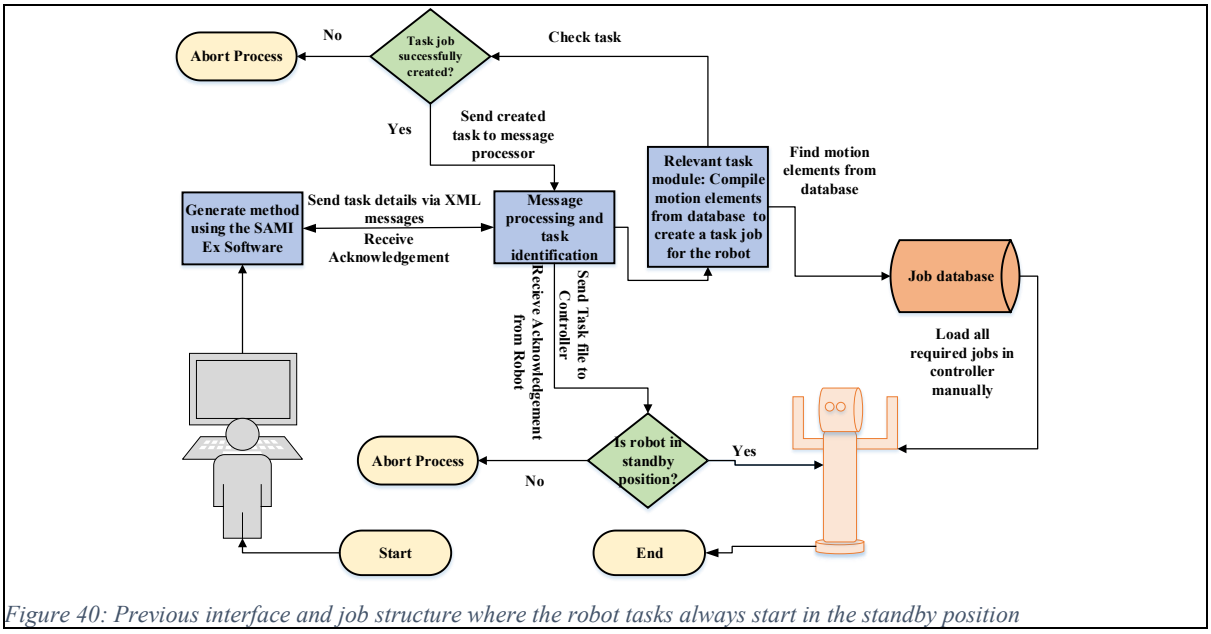


Figure 40: Previous interface and job structure where the robot tasks always start in the standby position

The design of motion elements is such that the robot can prepare only one batch of samples per process. Multiple consequent sample processing is not possible without manual intervention. The main reason for this limitation is the design of SAMI method. As discussed in the earlier section, multiple SAMI method files are queued in a defined sequence to create a sample preparation process flow. The labware positions are hard coded in the method files. In case of continuous sample preparation in multiple batches, the positions of usable labware

changes for each batch. Hence, the method file queue used for the first batch cannot be reused without programming changes. This limitation affects the productivity of the sample preparation system using the Yaskawa dual-arm robot. The automation system thus has scope for further advancement and optimization. Five different aims have been defined in order to enhance the current dual-arm robot platform.

1. **Time saving:** In RS-1 the robot requires nearly double the time to create one batch of samples as for laboratory technician. Although the accuracy is comparable, the extra time required poses a disadvantage. The goal is to reduce the overall time required to carry out individual tasks, as well as the process as a whole. This goal is achieved by re-planning the robot motions using a shorter route. However, the arm motion is limited by angular rotations of each joint. The two arms move simultaneously to reach the required position. Motions over a long distance, with compound arm rotations introduce shaking. Liquid contained in a labware may spill during motion, or excessive shaking of a heavy labware can cause damage to the fingers. Reducing the processing time is important as it directly affects the number of samples prepared per day. In addition, majority of the samples are prepared in highly volatile reagents such as hexane. Exposure to air leads to changes in concentration of the samples which affects the end measurement results. Hence, the motion path and speed have to be altered to reduce this effect. The speed of the motions is increased to the maximum possible value to achieve maximum speed with minimum shaking.

2. **SAMI EX software and process generation**: The time required for creating SAMI Ex methods is also significant. Previously, a single sample preparation process was designed in segments using the SAMI Ex software leading to creation of multiple method files for a single process. The large number of files was tedious to track and it gives rise to possible errors in continuity of the process in terms of labware placement. Hence, a new way of creating SAMI methods is introduced such that one method can be used continuously for multiple sample preparation runs. A single method file allows the operator to see the entire process in one glance and keep track of process edits and flow of labware. The new version of SAMI method files also allows a single file to be used for preparing multiple batches of samples.

3. **System flexibility:** Life science applications involve creating a variety of samples. Measurement samples frequently vary in composition, volumes and reagent properties. The batch sizes vary greatly ranging from a single sample to hundreds of samples. Multiple different samples have to be prepared in a day and each process may involve varying tasks. Automation of such a lab requires the system to cope up with frequent changes. Unlike in manufacturing plants, where a robotic system is dedicated to performing the same task for years, laboratory systems might need to accommodate multiple processes in a day. Hence, the system should be easily and quickly reconfigurable. The robot should adapt to any new process without major system changes or extra robot programming. The interface program should be able to accommodate new instruments without affecting the working of the existing system. Additionally, the robot motions created for current tasks should be reusable for other tasks as well. The robot should be able to transition from one task to another smoothly without too many positional adjustments. The system should be so planned that all the

available space and resources are efficiently utilized. The system should be programmed such that it can also be applied for bulk sample preparation in a commercial scenario. The programming of robot task jobs should be standardized so that any new user can easily comprehend and adapt to the system.

4. **Introduction of new instruments and labware:** Following the exact procedure is essential for successful sample preparation. Specialized automation tools might introduce some deviations in a process as compared to conventional lab instruments used for manual processing. Thus, the sample preparation needs to be done using the same tools and labware as used for manual processes. The end effector design of the robot is different than a five-finger human arm and the instruments used in a manual sample preparation process might not be directly compatible with the robot. A number of lab instruments such as the manual pipettes, LC tray etc. have hence been previously adapted for the robot using custom made fixtures. More such labware is newly programmed to be used by the robot. A special adaptor was designed and printed for using fragile glass pipettes with the robot. In addition, electronic pipettes were newly introduced in the system. The robot was programmed to interface with a standard solid phase extraction (SPE) module. Various labware can be transported to the SPE instrument and liquid can also be pipetted on the SPE by the robot. A new cooling rack was introduced on the robot workbench to hold derivatization reagents at the required temperature. A new gripper system with force feedback is installed. The new grippers detect the presence of labware during transportation. New gripper fingers and gripper mounting was designed as well.

5. **Controller space management:** As mentioned before, the robot currently uses a large number of motion elements to carry out the various tasks. However, the FS100 controller, capable of holding only 10,000 robot steps at any given time, is not enough to accommodate all the motion elements at once. The robot can only access and perform tasks that are stored in the controller memory. Hence, motion elements relevant to a process have to be manually copied to the controller before starting a process. This puts restrictions on the number of tasks that can be carried out and the variety of processes as well. The motion elements transfer cannot be carried out while a sample preparation process is going on. Manual transfer of motion elements also leads to errors if some relevant files are left out or a wrong version of the file is copied to the controller. In order to benefit from the entire range of robot capabilities, it is necessary that the robot can access all available motion elements. Hence, they have to be so redesigned so that the files are smaller in size than previous while carrying out all required motions. All the files should be stored to the controller at a time. The newest version of the motion elements should be automatically loaded to the robot controller before starting the sample processing. Reducing the total number of motion elements required for the entire system as well as their simplicity of understanding for a new user is also a priority.

5 Optimization Strategies

A number of approaches were considered to achieve all the goals listed above. The various approaches are discussed in the following sections. The various functionalities were implemented in the interface program. The optimized robot interface (optimized R-Interface) program has hence forth been termed as R-Interface 2.0.

5.1 Motion Planning Using Teach Pendant

The RS-1 uses the robot teach pendant to create motion elements in the Inform III language. The robot arm is moved along the desired trajectory using the teach pendant. Multiple points along the trajectory are saved in the form of joint angles of the robotic arm. The robot can retrace the trajectory with the help of the saved coordinate positions and arm posture angles. The other method for creating motion elements is using motion frames. Both the methods are discussed in detail in later chapters.

Motion elements created using the teach pendant ensure exact positioning of the end effector in order to manipulate labware. However, multiple such motion elements need to be created and tested which is time consuming. In order to reduce the programming effort involved in creating motion elements, it was considered to implement a combination of motion elements and direct arm manipulation via the R-Interface 2.0. The direct manipulation of robot arms can be used to change the arm posture to reach various areas of the robot environment directly by sending coordinate positions to the robot via the R-interface 2.0 without the need for motion elements.

The robot has a set of APIs for direct manipulation, where the relevant variables have to be set. The arm number to be moved, type of motion and motion speed have to be specified. In addition, the robot arm angles in the target position have to be defined as well. A separate database can be created for storing this data and recalled by the API as required. The robot controller has the ability to compute the path from its current position to the required position automatically. Such motions include moving the arm from the workbench to the shelf or the pipette stand, etc. Interference areas and obstruction areas can be defined in the robot environment so that they are avoided during motion. The robot can be allowed to move at high speeds during such transitions where precise positioning is not required.

The commands to move are sent to the robot controller via ethernet in the form of data packet requests which are composed of a 32-byte header part and a 479-byte data part. The data part holds information to robot arm positions and speeds (Figure 41).

Data part

32bit integer	Byte 0	Byte 1	Byte 2	Byte3
1	Specifying control group (Robot)			
2	Specifying control group (Station)			
3	Specifying the classification in speed			
4	Specifying a speed			
5	Robot 1st axis pulse value			
6	Robot 2nd axis pulse value			
7	Robot 3rd axis pulse value			
8	Robot 4th axis pulse value			
9	Robot 5thaxis pulse value			
10	Robot 6th axis pulse value			
11	Robot 7th axis pulse value			
12	Robot 8th axis pulse value			
13	Tool No. (0 to 63)			
14	Base 1st axis position (Pulse value)			
15	Base 2nd axis position (Pulse value)			
16	Base 3rdaxis position (Pulse value)			
17	Station 1st axis position (pulse value)			
18	Station 2nd axis position (pulse value)			
19	Station 3rdaxis position (pulse value)			

Specify robot number

Specify motion type

Specify motion speed

Specify arm joint values at target position

Specify base axis values at target position

Figure 41: API for moving the robot arm directly via the interface program

In combination, motion elements can be used for finer and more precise motions such as picking or placing labware, individual vials or interacting with instruments. Such a combination of motions can be useful for carrying out various tasks consecutively in a random order adding to the flexibility of creating a large number of processes with the existing system setup (Figure 42).

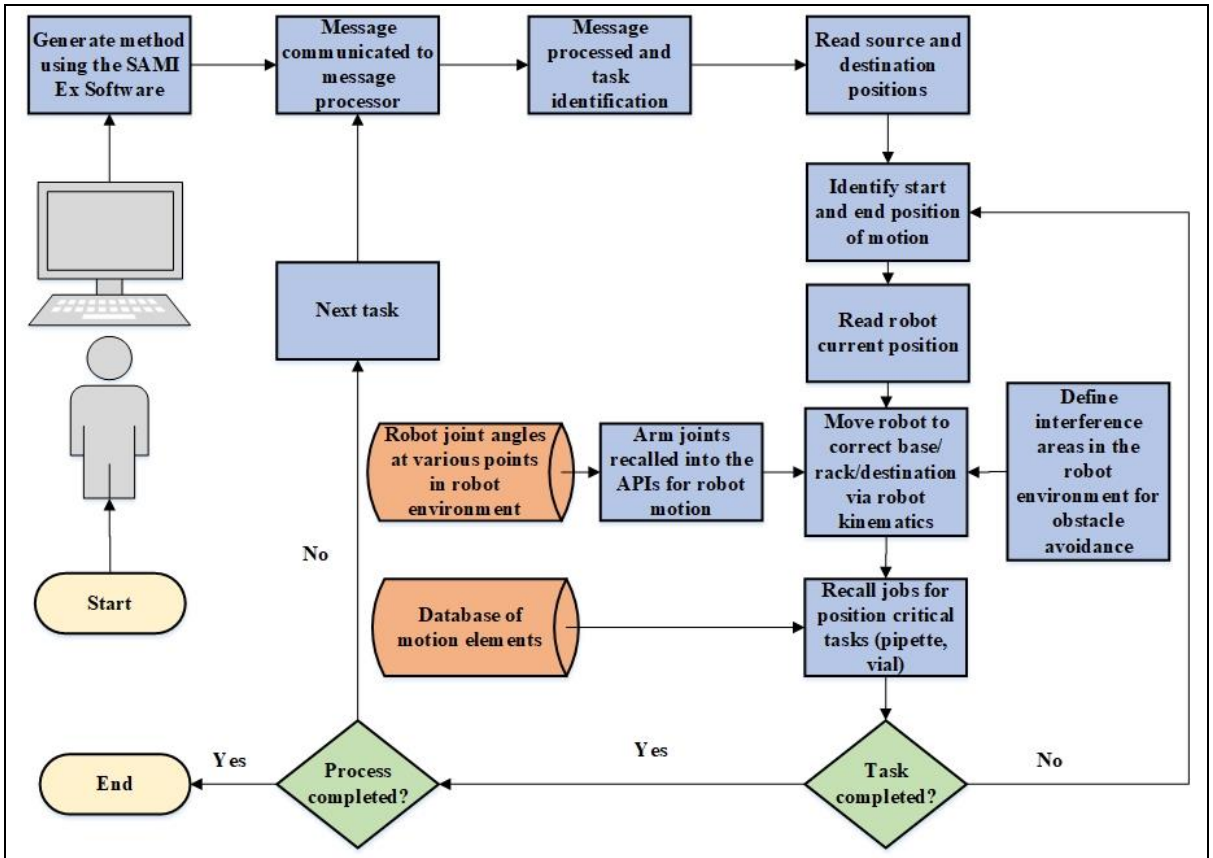


Figure 42: Utilizing robot kinematics to compute the motion without a fixed start and end point for a task

5.2 Path Planning Algorithms

Robot navigation consists of four basic components: perception of environment, determination of location, path planning and the regulation and control of motion. Path planning algorithms can be broadly distinguished into classical algorithms and heuristic algorithms (Figure 43).

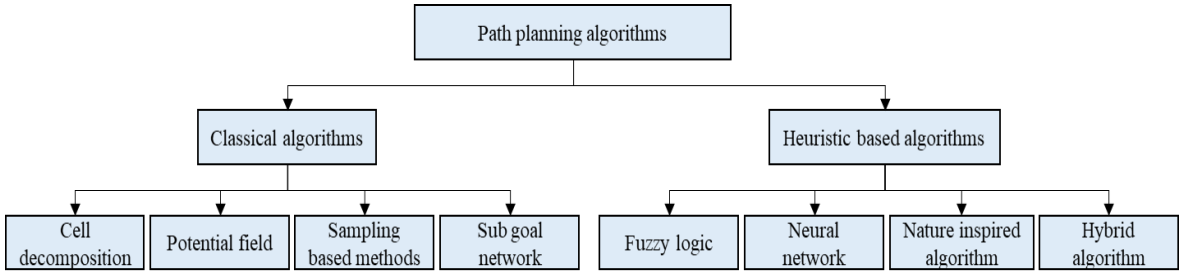


Figure 43: Classification of path planning algorithms

The classical method of cell decomposition involves configuration of the free space in the robot environment into small regions called cells. The free cells are used to identify a path to reach the target. The obstacles and goals are assigned attractive and repulsive forces in the potential field method of path planning. Sampling based motion planning has been most widely used due to their capability of solving real world problems. The rapidly-exploring random tree (RRT) algorithm is an example of sample-based planning. The RRT algorithm takes the current point as the start point and the destination point as the end point. The algorithm tries to find the

path between these two points automatically by searching its immediate surroundings in small iterations and creating a tree with branches linking random points. If the direction of the motion is closer to the destination the algorithm keeps searching further, if the distance is increased, path search in that direction is terminated. The RRT* algorithm was developed as an improved version of the RRT and is claimed to guarantee an optimal path. The A* and derivative algorithms of A* predict a path with the least cost function from the source point to the goal point. The cost function is the error between the actual value and the predicted value [113],[114],[90], [115]. Classical path planning algorithms are usually preferred in many real-time motion planning algorithms.

Heuristic algorithms provide can approximate solution to problems but do not guarantee an optimal solution. Complicated mathematical models are generated depending upon the kinematic and dynamic properties of the robot arm joints, the various forces acting on the arms as well as regulation inputs from various sensors, if any, have to be included in these models. These mathematical models are then applied to find the short path or the path with least resistance for the robot to move. Neural networks are based on a collection of nodes which are connected to each other and can transmit signals to each other. The path learning algorithm of a neural network may not guarantee an optimal solution. Fuzzy logic algorithms are created to develop the ability of the robot to navigate without exact calculations. The decision making in fuzzy logic is commanded by a set of IF-THEN rules. Heuristic algorithms are more intelligent compare to classical methods. They can adapt to uncertain and constantly changing environments and are often applied on autonomous navigation systems [93], [116].

Path planning algorithms have advantages such that the path is calculated automatically. Obstacles can be mapped in the robot surroundings and the robot can work around obstacles. The motions are created in real time and hence need not be saved. Hence, manual teaching is not required. Hard coding the robotic arm movement can also be avoided giving the system freedom to carry out any processes in the future.

5.3 Offline Motion Planning

A number of software are available for the 3D simulation of a robot environment. Robot Operating System (ROS) is one such open-source programming framework aimed to support code reuse and simplify building and running robots. The ROS is applicable to a vast number of robotic systems including, mobile robots, aerial robots and industrial robots. Commercial industrial robots have been traditionally installed to perform highly specialized tasks and the underlying operating software were also built with the same intention. Each system was different and lacked standardization leading to long development times. The ROS framework facilitates the sharing of technical knowhow about building and controlling robots reducing the efforts in setting up a new robotic system. Different robots have different requirements, and it is difficult to find one standard solution for all problems. The open-source design of ROS helps to overcome this obstacle. ROS provides services such as hardware abstraction, low level device control and implementation for commonly used functionality, message passing between

processes and package management. It provides tools and libraries for obtaining, building and writing code across multiple computers.

Common challenges such as coordinate system transformation, motion planning, sensor integration, etc. have field tested ROS solutions that are ready as plug-ins thus providing a modular software support. The robot programmer only needs to download the relevant software from the ROS library and include them as part of his robot system design. ROS provides a runtime environment supporting real time communications between system environment and data sharing. Specially designed development tools are available that help monitoring, troubleshooting and visualizing a robot which can be used to develop a robot simulation and fine tune the motions. Other complementary open-source software such as MoveIt, Gazebo, OpenCV and Microsoft .Net Micro framework can be implemented with ROS to create a comprehensive robotic control system. MoveIt provides a high-quality software for motion planning and manipulation of objects in 3D environment. Gazebo allows the user to create visual simulations of multiple robots in real-world environment. OpenCV is an open-source library supporting computer vision for analyzing real-time videos. The Microsoft Micro framework can be implemented to create applications electronic board level devices (Figure 44).

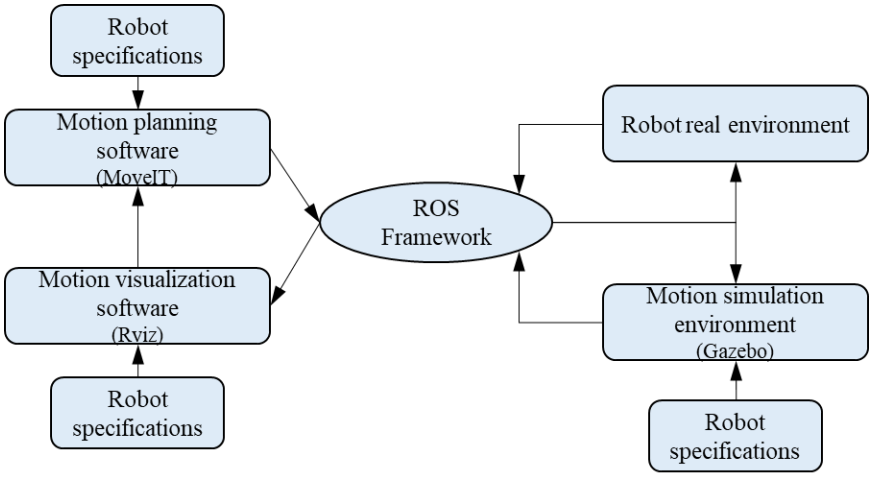


Figure 44: High level diagram of a robot controlled using ROS framework in collaboration with compatible third-party open-source software

The ROS model consists of a master, nodes and topics. Nodes are single processes with a unique name and form the backbone of the ROS framework. They are pieces of software and can be written in any programming language. Nodes can do small tasks like read a sensor or control a server. Topics are named buses over which nodes can exchange information. The nodes communicate with a publish-subscribe protocol. If a node has information to share, it has to publish to said topic in order to be able to send information. If another node is interested in the information, it needs to subscribe to the said topic in order to access the information. Nodes may advertise or call services from one another, where services are calls to a function that is executed in another node (Figure 45). Such an offline process can be used to preplan the most optimum robot motions before implementing them in real environment. The robot thus has a significantly less down time during the development phase.

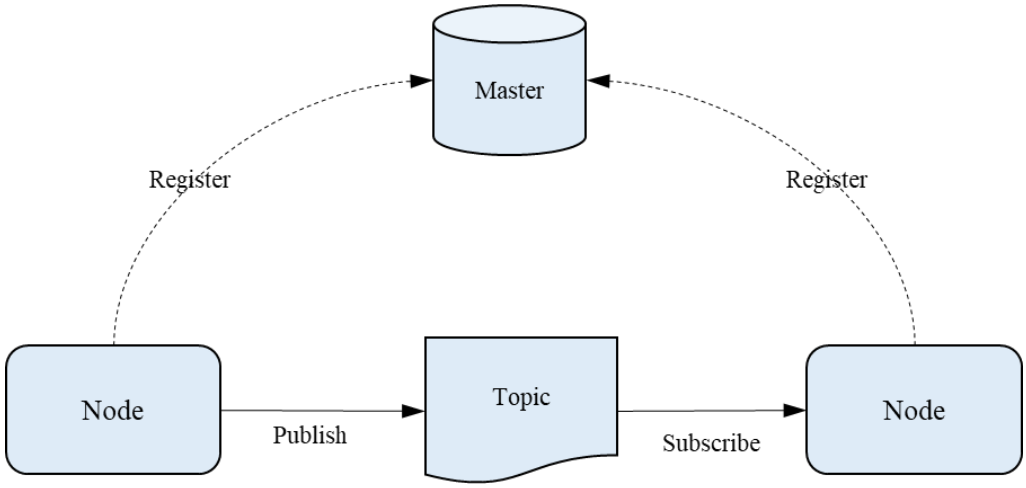


Figure 45: Overview of the communication paths within ROS framework

5.4 Comparison of Possible Approaches

In order to finalize the best suited approach to carry out the modification and optimization of RS-1, the advantages and limitations of the three earlier described approaches were identified and compared. In case of the first approach of combined direct manipulation and motion planning, it was seen that the robot could not independently perform a complex motion that required rotating all the joint angles at the same time. The motion had to be simplified by the programmer by defining intermediary positions. Thus, the robot arm posture transformation was carried out in multiple steps. Thus, a large database of robot positions with corresponding joint angles will have to be created that can be input to the API. This requirement conflicted with the original intention of reducing the number of fixed paths and teaching efforts. Moreover, the motions defined via such API's cannot be tested manually with the teach pendant. Hence, the arm motion path was unknown and unpredictable during the motion development and planning phase. Multiple interference zones had to be defined to avoid collisions. Interference zone is a cubic region of the workspace that the robot tool center point is not allowed to enter. Shelves, hotel structures as well as lab instruments would have to be defined as interference zones to avoid free motion of the robot in these areas during direct manipulation. However, the robot is required to access these areas intermittently using motion elements to access labware, press buttons, open lids to the instrument etc., during which time the corresponding interference zone would have to be disabled to allow motion. The programming of such a system is complex and can make it unreliable in terms of safety.

Path planning algorithms provide numerous methods to program the robot motions. However, they have a learning phase which is necessary for optimum results. They also require long computational times. They require precise information about the robot environment and often about the robot kinematics and dynamics as well which is not readily available. The robots work in unpredictable environments, hence outside forces affecting the arm motion cannot be accurately estimated. If the robot is in an unusual posture, or if the destination has multiple paths available or if there are a number of obstacles in the robot environment and the

appropriate heuristic method is not applied, the algorithm may reach a dead end or go back and forth between two points without reaching a solution. Any change in the robot environment, an added instrument, added base, etc. will have to be carefully mapped in the system. The operator does not have any control over the motion of the robot during runtime. Hence, motions need to be visualized in a virtual environment before implementation. Moreover, the approach of implementation of path planning algorithm has nothing in common with the existing system RS-1. The progress done with the system until now will be either totally or partially lost. Overall, the approach will prove to be extremely time consuming.

The RS-1 is currently implemented on a Microsoft Windows platform. The .Net framework was selected as the preferred technology due to its ability to support a variety of coding languages which is an advantage while creating interfaces with the various lab instruments such as the GC, LC and the SPE. At the time of conceptualization of this project, the ROS framework was only available for Unix based platforms. Thus, introducing ROS for planning the robot tasks involved overhauling the entire system making all previous advancements redundant. The offline mapping system also needs precise mapping of robot environments and actual dimensions of the robot.

The three approaches were compared qualitatively on the basis of five major factors.

- **System flexibility**: ability of the system to carry out various tasks in a randomized order
- **User control**: ability of robot operator to stop or control motion during the developmental phase
- **Development time**: time required to set up a fully implemented running robotic system
- **New implementation**: introduction of new sample preparation processes without changes to the system
- **Teaching effort**: Effort of teaching new motions to the robot

Table 3: Qualitative comparison of the three path planning approaches

	Direct manipulation	Path planning algorithm	Offline path planning
System flexibility	Moderate	High	High
User control	Moderate	Low	High
Development time	Moderate	High	High
New implementation	Moderate	Moderate	High
Teaching effort	Low	Low	Moderate

5.5 Final Approach

The RS-2 needs to be conducive and robust for uninterrupted sample preparation and addition of new tasks in the future. It also needs to be comprehensible to a new user. Based on the characteristics of all the three approaches as summarized in Table 3, the advantages and limitations of all the approaches were considered. The approach of path planning allows the robot to be moved to an arbitrary position without any teaching effort. However, this approach generated safety concerns as the motion of the robot arms is unknown and can cause collisions.

In order to avoid collisions, a high number of constraints (speed, postures, allowable paths, etc.) will have to be applied to the path planning algorithm. The constraints might change with minor changes to the robot environment. The cost and effort of reconfiguration of the algorithm is high and time consuming which is incongruent with the goal of the dissertation. The approach of offline programming offers a number of advantages such as minimum downtime of the robot and flexible motion planning. However, this approach requires a change in the operating software from a Windows platform to Linux platform. The development of the R-Interface done until now might get invalidated. Due to these observations, it was decided to adopt the first method of job teaching (as done in RS-1) with slight changes. The robot will not be moved via direct manipulation. In order to reduce the total jobs and save controller space, the structure of the motion elements will be changed, so that multiple motion elements are clubbed together in a single file. All new motions would be programmed using the teach pendant.

In the RS-1, multiple motion elements are created to cater to the variety of labware. The motion elements for a task may differ in gripping posture, gripping position or gripping force. However, the basic transfer path is common. To reduce the number of motion elements, the different motion elements will be combined in one file in the RS-2. For jobs involving labware gripping, the common path will be taught. Customized gripper positions will be incorporated in the same motion element using variable indices that will be set automatically via the R-Interface 2.0 based on the labware type. The robot needs to transfer labware to multiple destinations on the workbench. Hence, for transfer jobs, one basic motion would be taught and the motion will be mapped for all the ALP positions using reference points. In the future, if a new labware or ALP position is introduced, a new variable index or ALP reference point can be added in the existing motion element. This action will not affect the previous programming and would also solve the problem of having too many motion element files to handle.

Each ALP on the workbench will have two reference points instead of one: one for each arm. Both the arms will be fitted with different end effectors, carryout different tasks, and achieve different postures. Hence, the reference points for both are not interchangeable. Some motion elements using both arms use the relative path planning method (discussed in the further chapters in detail). To synchronize the motions, relative distances between the arms will be calculated and arithmetic calculations will be used to find final arm postures. It will also be possible to adjust one arm independent of the other in case of future changes. Figure 46 shows an example of a motion element file used to load pipette tips to a manual pipette. Four of the six single channel manual pipettes require identical motions for loading tips. Instead of creating an individual motion element for each of the four pipettes as done in RS-1, only one motion file is created in RS-2 to carry out the same task.

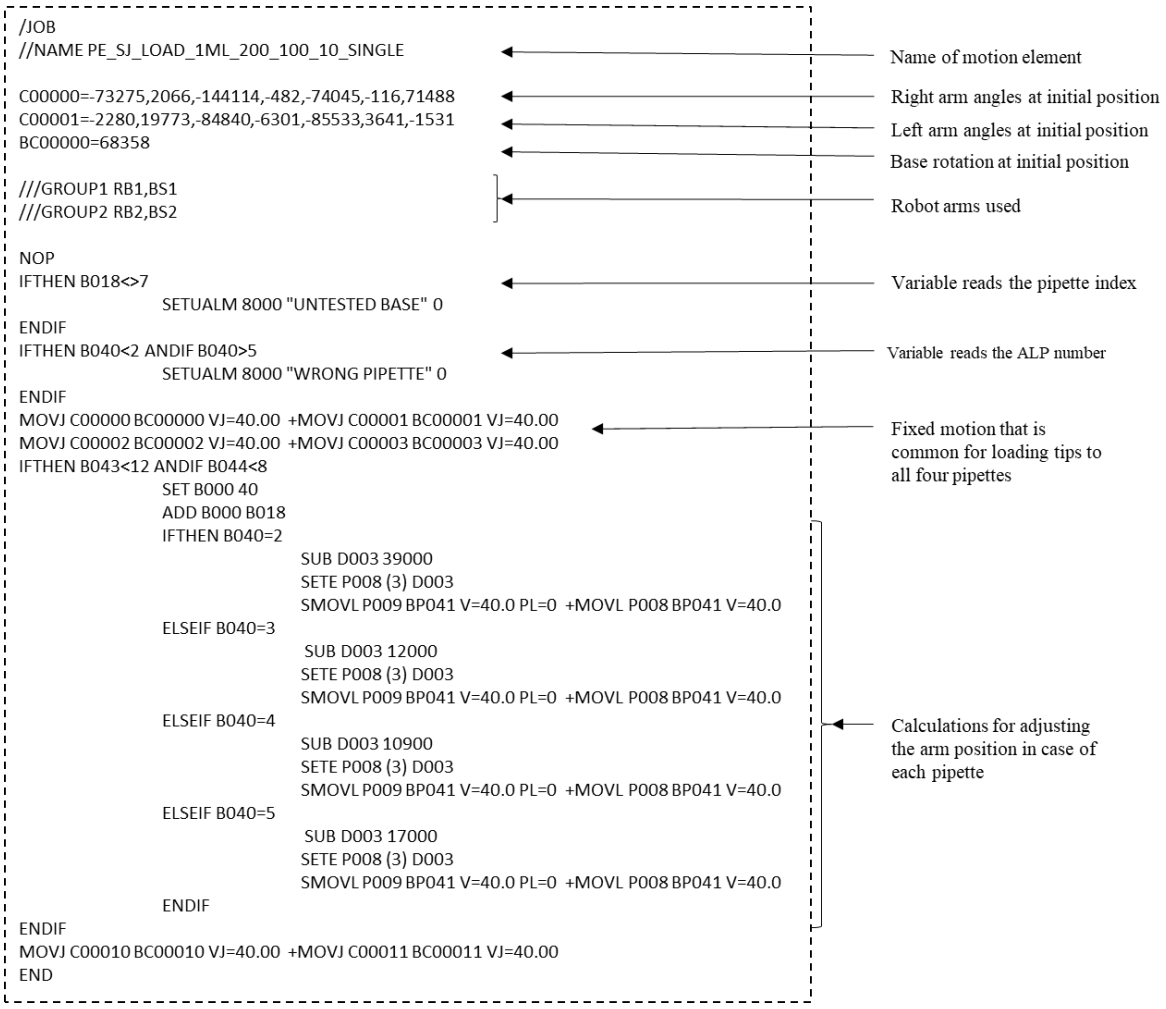


Figure 46: Example job for loading tips to pipette

All motion elements required would be transferred automatically to the controller before the start of a process. These files would be deleted from the controller at the end of the process. Speeds of all motion commands in the motion elements would be increased in order to make the system faster and save time. The default position of the robot will also be changed. The arms will be repositioned in a posture such that the time required for approaching various areas in the robot environment is reduced.

The structure of the R-interface would be changed such that the flow of the program and commands is streamlined. Every individual task, such as labware transport between various positions, liquid transfer, etc. will be handled by completely individual modules and classes. All the modules will only interact with the central message processing code. The modules would not be dependent upon each other. This way it would be easy to add new or delete existing functionalities in the R-Interface (Figure 47). Programming changes to one module will not affect the others.

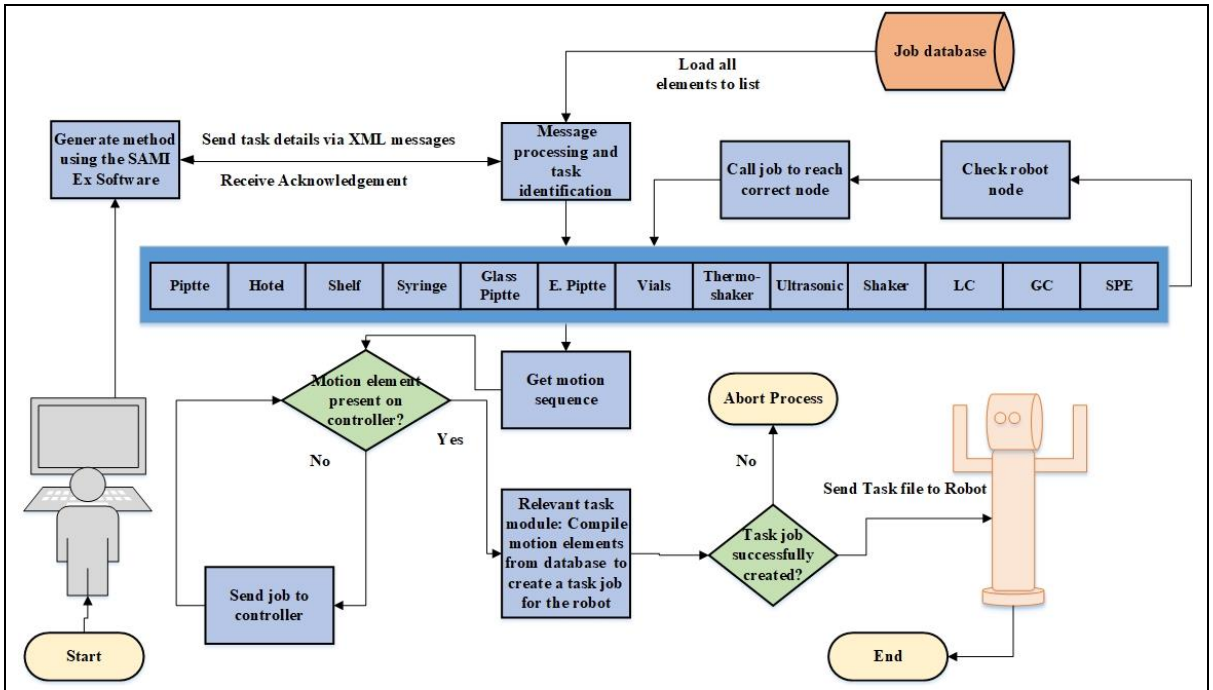


Figure 47: Proposed process flow for flexible and fast sample preparation

With this method of robot programming, the programmer can control all functionalities of the robot. The motions can be planned as required and can be easily edited in the future if necessary. The robot can also be tested manually multiple times before it is set to work independently to eliminate any possibilities of collision or unsafe motion. New tasks or instruments added to the work environment do not affect the existing robot motions.

6 New Technology Introductions

In order to aid the optimization of the system and expand the capabilities of the existing RS-1, new features have been introduced in the RS-2. New instruments such as the SPE, a cooling rack and electronic pipettes are integrated with the robot system. The end effectors have been replaced to extend features.

6.1 Electronic Pipettes

Complex sample preparation procedures often include multiple liquid transfers. The volumes on the nine manual pipettes available in the RS-1, pre-set manually to a fixed value at the beginning of a process, cannot be reset by the robot. Thus, for long processes, the volumes of each pipette have to be meticulously planned so that pipette combinations can be used to deliver all the required reagents in correct quantities. An example is presented with the help of Table 4. A process of preparing cholesterol calibration samples in 2 mL vials has been considered for the example. Three samples each of 10 ppm, 5 ppm, 1 ppm, and 0.5 ppm solutions have to be prepared in a single process. Hexane, 5-α cholestane and cholesterol have to be added in various volumes as can be seen in the table. Only four single channel manual pipettes (10 µL – 1,000 µL) are compatible in size with 2 mL vials. Multi-channel pipettes are incompatible with vials. The various reagent volumes required to create these samples are given in the table. Volumes of each of the four pipettes are set accordingly such that one reagent can be dispensed in multiple steps using multiple pipettes.

Table 4: Depiction of the various liquid volumes dispensed using a combination of four single channel manual pipettes set to a fixed value

Volumes set in each pipette				**Volumes of various reagents required during the entire process**
1 mL pipette	**200 µL pipette**	**100 µL pipette**	**10 µL pipette**	
800 µL	+ 180 µL		+ 10 µL	990 µL
800 µL	+ 180 µL			980 µL
800 µL		+ 100 µL		900 µL
800 µL				800 µL
		2 x 100 µL		200 µL
		100 µL		100 µL
			2 x 10 µL	20 µL
			10 µL	10 µL

However, combining multiple pipettes is not possible in all cases. For example, if a volume of 182 µL were required in the above example, an additional pipette might be required, or the pipetting steps will have to be broken down further into smaller volumes. Besides, frequently changing pipettes is time consuming. Hence, two electronic pipettes, with a volume range of 5 µL – 200 µL and 50 µL – 1,000 µL respectively are introduced in RS-2 and are hung on a rack in front of the robot (Figure 48).

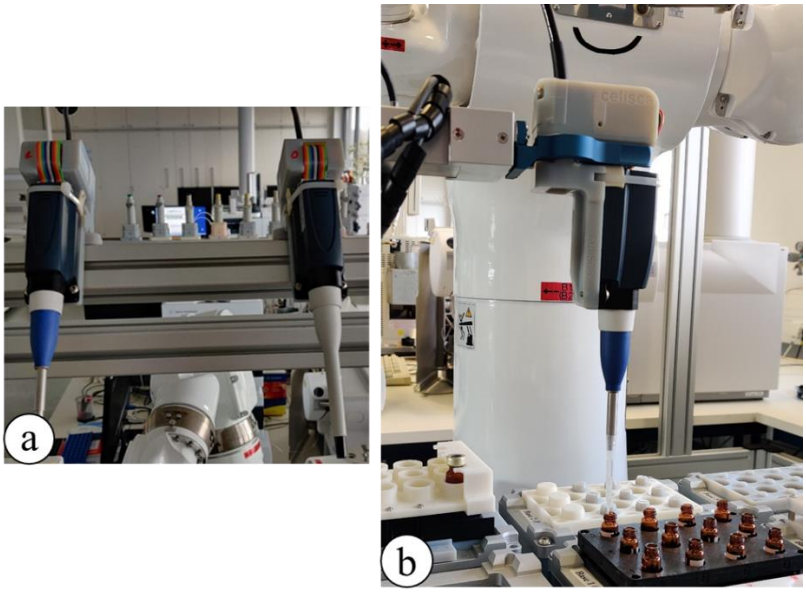


Figure 48: (a) Electronic pipettes hanging on the overhead shelf, (b): single arm operation of pipetting liquid into a vial

A hardware interface box, allowing the operation of four pipettes, was developed and is connected to the controlling computer via USB. The electronic pipettes are connected via RS232 and identified using virtual COM ports [38]. The liquid volumes on the pipettes, as well as the aspiration and dispensing actions are software controlled. An alternating software and R-Interface 2.0 control is implemented to carry out the liquid transfer. The pipetting commands are integrated into the robot motion sequence in the R-Interface 2.0 to alternately call the motion elements and pipetting commands.

The RS232 communication code is written in Visual Basic (Microsoft, Redmond, USA). The pipettes are configured using messages in the 7-bit ASCII code format. The messages contain six different components which includes commands and data to carry out specific actions. Several commands are available for the initialization and pipetting (Table 5). A seven-segment angular sensor on the electromechanical piston provides position feedback and has to be initialized by moving to all possible positions before pipetting. Aspiration speeds and volumes are set using the pipetting commands.

Table 5: Commands and actions implemented by the electronic pipettes

Commands	Action	Value Range
Configuration	Communication between controlling PC and pipette	
Initialization	Sensor initialization	
Stepwise retraction from current position	Aspirate set volume of liquid	5 µL – 200 µL 50 µL- 1,000 µL
Stepwise extension from current position	Dispense set volume of liquid	5 µL – 200 µL 50 µL- 1,000 µL
Selection of retraction speed	Set speed of aspiration	1 - 6
Selection of extension speed	Set speed of dispensing	1 - 6
Tip-drop and move to end position	Tip reject	

A graphical user interface rLine_Com was created to test the communication and commands to the pipettes. At the beginning of the application, all available COM ports are displayed. By double clicking the available port, commands to the particular pipette are implemented. The parameters are specified by entering values in text fields or moving track bars. Commands are executed step wise with the click of a button.

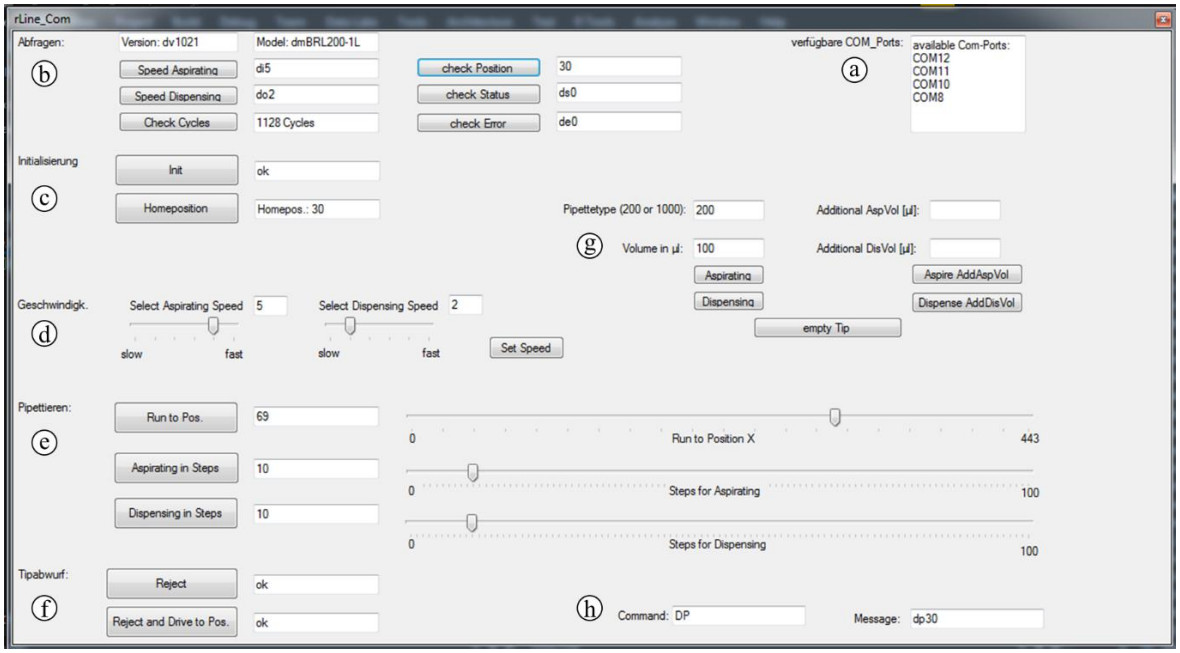


Figure 49: (a) available COM ports, (b) query pipette properties, (c) initialize pipette (d) set speed, (e) set piston position, (f) check tip eject, (g) set volume of liquid, (h) command sent and reply received [38]

The advantage of this instrument is that a single pipette can be used for a range of volumes saving valuable space required for storing labware. Significant time required to transport the pipettes is also saved. The liquid transfer task is not limited by the number of pipettes available. The robot utilizes only the right arm for the task. The motions of loading tips, dispensing and aspiration of liquid are identical to the right-hand motions of a manual pipette task. The left arm remains unused and maybe used to carry out another task synchronously. The left arm may also be programmed to perform another pipetting task in parallel.

6.2 Gripper System

Figure 50: Previous end effectors (grippers) without force sensing

RS-1 implemented the LEHF20K2-48-R86P5 end-effectors with LECP6 controller (SMC Corporation, Tokyo, Japan). The fingers of both the end-effectors are 3D printed to a custom design and differ in shape. The left arm fingers are designed to transport plate type labware such as MTPs, pipette tip boxes, vial racks, etc. Fingers of the right arm are designed to handle delicate labware such as glass vials, glass syringes, vial lids, pipettes as well as to operate push and turn buttons.

The grip width of the fingers changes with the labware and labware shape. For example, the MTP lid is wider than an MTP and hence, needs a different width adjustment.

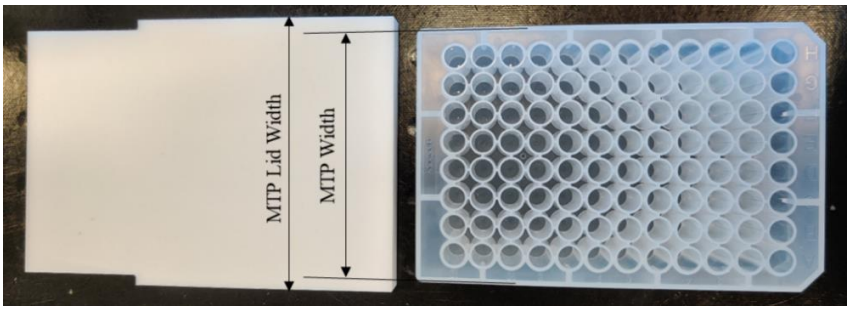


Figure 51: Width difference in MTP and MTP lid requiring different grip settings

In order to grip an object, the gripper fingers have to move to a certain position according to the size of the labware. Table 6a below lists positions required for some of the commonly used labware. The grip positions had to be decided and accurately programmed by the programmer in RS-1. This was done using a trial-and-error method. After the introduction of a new labware, in case a previously programmed grip width was not compatible, new grip width had to be programmed.

Table 6: Right and left end effector grip widths for various labware

Right gripper		Left gripper	
Object / task	**Position (unit: mm)**	**Object / task**	**Position (unit: mm)**
Open	1.0	Open	1.0
close	48.0	close	48.0
4 mL vial body	6.1	4 mL vial body	7.2
4 mL vial cap	7.7	10 mL vial body	11.7
Glass pipette	1.6	MTP	22.9
Pipette	24.6	MTP lid	44.2
Thermoshaker lid	3.4	Tip box	39.1

(a)

Step No	Move M	Speed	Position	Accel	Moving Force
		(in / s)	(in)	(in / s2)	%
0	Absolute	2.0	0.039	78.7	150
1	Absolute	1.0	1.100	78.7	150
2	Absolute	1.0	0.670	78.7	150

(b)

Gripper control comprises of two parts: setting finger positions (also called steps) and creating gripper jobs (Table 6b). A list of maximum 64 steps is mapped to various finger positions, speed of movement of fingers, etc. The gripper controller communicates with the FS100 by sending the step number via a 6 bit I/O port. Individual robot jobs are created for each step using the FS100 teach pendant. The end-effector drive is turned on and the fingers are moved to the corresponding position. An alarm is raised if the fingers do not reach the designated position.

The gripper system cannot recognize if an object had been held or not. A slight angular deviation was observed after dismounting and re-mounting the end-effectors due to lack of positioning pins. Moreover, the 3D printed fingers showed slight deformation over time. This led to error in gripping action of 2 mL vials and vial lids.

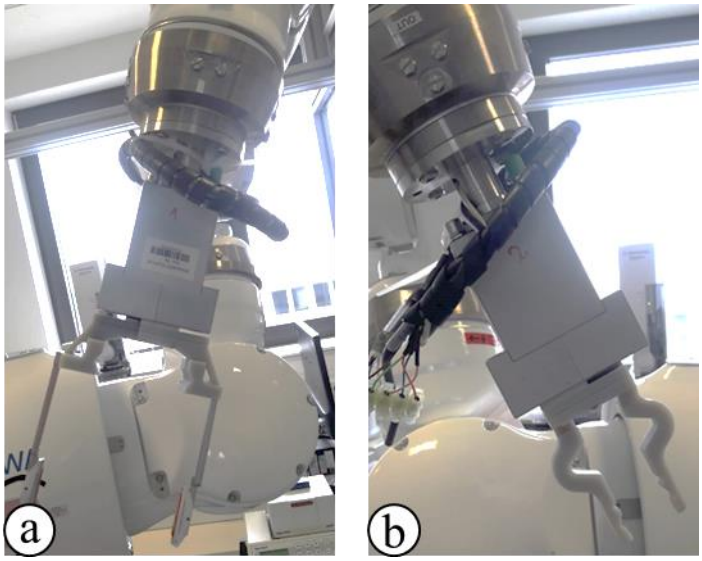


Figure 52: New gripper fingers with force sensing

To overcome the above-mentioned drawbacks, the previous end-effectors were replaced in RS-2 with new grippers allowing force feedback (Figure 52). The SG0150 servo end-effectors are manufactured by PTM Production, Egenhofen, Germany. With the help of these grippers, robot can transfer objects from one hand to the other. It can perform humanly tasks such as turning knobs, pressing buttons, screwing and unscrewing vial lids, handling delicate glassware and even pinching actions as required for glass pipettes. Hence, the robot can be used to automate difficult sample preparation processes.

The gripper controllers communicate with the FS100 via an RS232 port. A software development kit (sdk) called MOTO GSI is available for this purpose. The MOTO GSI provides an INFORM instruction set extension to facilitate the establishment of socket connections for RS232 communication. The two gripper controllers communicate with a single port on the FS100 with the help of a diode that streamlines the communication messages. Thus, although both the grippers are connected, only one gripper can communicate with the FS100 at a time. In order to actuate the grippers during robot motion, three robot jobs are required. The ‘grip’ job, requires three arguments: gripper identification, grip width and gripping force. The grip width is always set narrower than the labware width. The grip job is used for gripping

an object. The gripper can exert a user defined force in the range of 3 N – 50 N. The gripping force is decided by trial and error based on the labware material, size and weight. The fingers exert a force on the object gripped. If an object is held between the fingers, the fingers receive an equal and opposite force of resistance which is detected by the gripper motor. That way the end effector knows if the gripper action was successful. The 'position' job requires two arguments: gripper identification and target position. The position job is used to open gripper to a desired width for picking object or to release a gripped object. A 'check' job is called after every grip job. This job ensures that the desired action is completed. If there is no opposing force, or the fingers reach the grip width, the robot operator receives an error message signifying the absence of labware. Depending upon the mode of the robot (teach mode or remote mode), a dialogue box appears either on the teach pendant or the controlling PC. The operator can then choose to retry the action, continue the motion or abort the task. Figure 53 depicts a snippet from a motion element created to transfer glass vials from the right arm to left arm. The grip job is executed first by providing relevant arguments. Check job is implemented to ensure gripping action is complete. The position job is executed next, to release vial from right arm. Check job is again executed to ensure the gripper fingers have reached target position. The robot arm is then ready to accept further commands

```
MOVJ C00000 BC00000 VJ=10.00  +MOVJ C00001 BC00001 VJ=10.00
MOVJ C00002 BC00002 VJ=80.00  +MOVJ C00003 BC00003 VJ=80.00
MOVJ C00004 BC00004 VJ=80.00  +MOVJ C00005 BC00005 VJ=80.00
MOVJ C00006 BC00006 VJ=80.00  +MOVJ C00007 BC00007 VJ=80.00
MOVL C00008 BC00008 V=100.0  +MOVL C00009 BC00009 V=100.0
CALL JOB:2_GRIP ARGF1 ARGF0 ARGF10
CALL JOB:2_CHECK1 ARGF1
TIMER T=0.500
CALL JOB:2_POSITION ARGF2 ARGF300
CALL JOB:2_CHECK1 ARGF2
```

Commands to move robot arms in position

Commands to grip vial with left arm
Arg 1: arm number, Arg 2: grip width, Arg 3: gripping force

Commands to release vial from right arm
Arg 1: arm number, Arg 2: target position

Figure 53: Example of motion element implementing the new force feedback grippers

Similar to RS-1, the end-effector system cannot detect if the labware fell out of the grip during motion. Hence, the grip job is called once again at the destination. As mentioned earlier, previously described gripping action and check job can check if the labware is still present or else raise an alarm. Unlike RS-1, it is not necessary to create new job files for different labware and gripping postures. A fourth job, 'speed set' job has also been created for optional use which can set the jaw speed of the position and grip commands. The grippers have a speed range of 3 mm/sec to 150 mm/sec.

6.3 Positive Pressure Unit for Solid Phase Extraction (SPE)

With a view to increase the capabilities of the robot, an SPE module was introduced on the workbench. Accurate determination of trace amount analytes is difficult without pre-treatment of the solution. In majority of the solutions, the concentration of the analytes is around the detection limit of the analytical instruments [117]. The solid phase extraction (SPE) module is primarily utilized to remove impurities and interfering biological matrix components in order to concentrate the targeted analytes. In order to expand the capabilities of the system, an SPE was introduced into the RS-2 and was placed perpendicular to the robot on its left side.

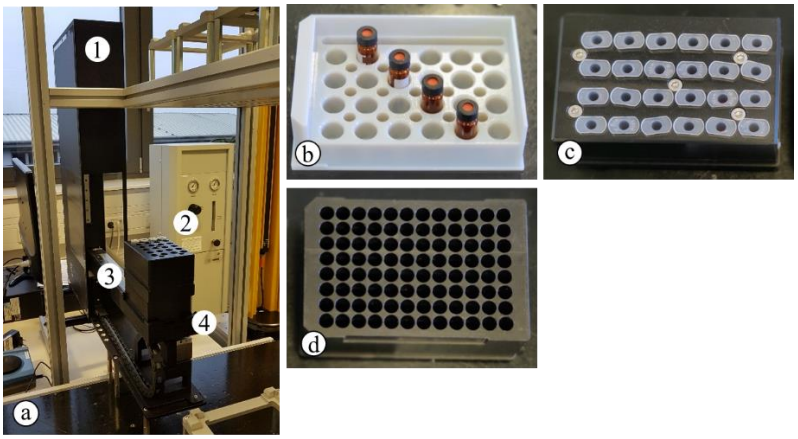


Figure 54:(a): SPE instrument with (1) SPE hood, (2) cartridge, (3) rinsing block, (4) carriage, (b) Specially designed 2 mL vial block with capacity for 24 vials, (c) cartridge adaptor, (d) rinsing block

The robot transfers the rinsing block and cartridges one by one from the hotel and the shelf to the SPE carriage. The two labware are placed on top of each other. Only the left arm is utilized for this task. The motion has to be very precisely programmed, to avoid collision with the overhead shelf, hotel structures, thermoshaker or the SPE itself. The robot is programmed to pipette directly from a vial on the ALPs to the cartridges using the electronic pipettes and 1 mL manual pipette. After liquid transfer, the carriage of the SPE is actuated directly through the SILAS module and is moved inside the hood of the SPE. Next, positive pressure is applied onto the cartridges to ensure and assist a uniform rate of liquid flow. In addition, a vacuum pump attached to the bottom of the carriage also assists in maintaining the downward flow of the liquid through the cartridges. The liquid transfer and pressure/vacuum application are done alternately for all steps of the sample concentration process. Before the final step of elution of the target analyte, the cartridges are put back to the shelf temporarily, the rinsing block is replaced with a custom designed vial rack. This rack is of 6 x 4 configuration against the standard 4 x 3 and sits perfectly under the 24-cell cartridge block. The cartridges are placed back on the SPE over the vial block, liquid is pipetted in the cartridges and positive pressure and vacuum is applied to the cartridges. The eluted target analyte is collected in the vials. The used cartridges are transferred back to the shelf while the vial rack is transferred to the ALPs. Further processing if required is performed on the workbench. The vials are then lidded and transferred to the GC-MS or the LC-MS autosampler for analysis.

6.4 Glass Syringe

The application implemented on the Yaskawa SDA10 robot requires derivatization, which enhances the measurement results after chromatographic analysis. The derivatization task has

been newly introduced in the RS-2. The derivatizing agent used is a highly toxic substance. Hence, it has to be stored in septum-protected vials and is placed on a cooling rack on the workbench during the sample preparation process. The cooling rack is attached to a pump and maintains the temperature of the reagent at 4°C to stop it from crystallizing. Hence, it can only be aspirated with syringe needles as the septum cannot be penetrated by a standard pipette tip. A reusable glass syringe is utilized to dispense precise and small quantities (10 µL – 200 µL) of derivatizing agent. These syringes (Hamilton, Reno, Nevada, USA) have an accuracy of ±1% of total volume. The robot is currently programmed to use a 25 µL and a 200 µL syringe. These syringes are fragile and expensive and hence have to be adapted for robotic use and meticulously programmed.

In order for the robot to be able to pick the syringe, a 3D printed adaptor was designed in-house. The syringe can thus be hung on a rack similar to the manual pipettes and is placed on the temporary pipette stand during the sample-preparation process. Figure 55a shows the prototype of the adaptor. The prototype had too many parts that had to be taken apart each time for changing the syringe which was time consuming. Besides, the sliding plunger experienced sweating and friction with the enclosure. The robot action was not smooth resulting in uneven forces causing the plunger or the robot finger to break. Hence a new design was developed as seen in Figure 55b. The new adaptor design is sturdier than the prototype. The plunger is guided by parallel steel rods resulting in smooth motion. The syringe can be changed by removing two placers situated in the front of the adaptor without having to dismantle the entire adaptor body.

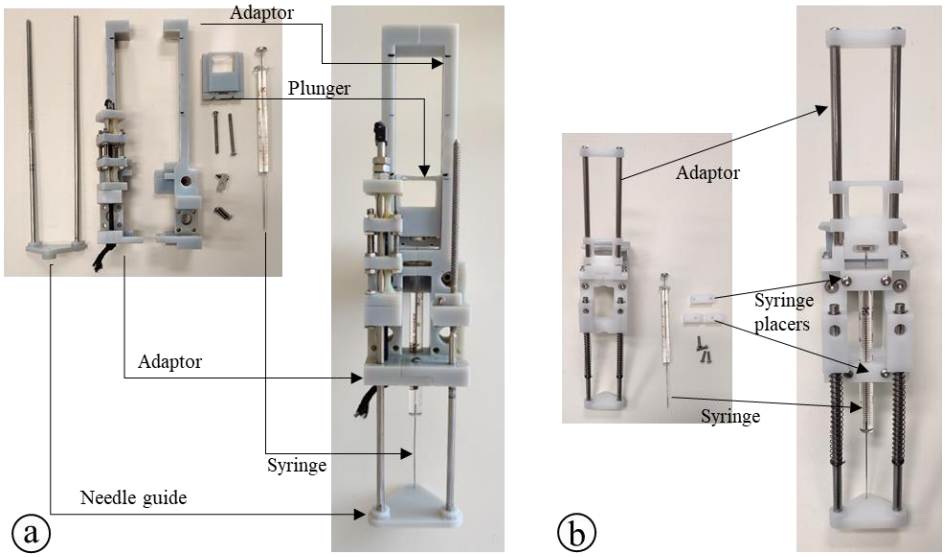


Figure 55: Glass syringe mounted into the holder (a): adaptor prototype during syringe change, (b): final adaptor design during syringe change

Both the robot arms are used for this task. The right arm picks up the syringe from the rack and moves to the cooling rack. The left arm actuates the syringe by pulling up / pushing down the plunger to aspirate or dispense liquid (see Figure 56). Before the reagent is aspirated, the robot washes the syringe with the help of non-reacting liquid to flush previous impurities from the bore, prevent cross-contamination or prevent syringe blocking. In case of the experimental application implemented in this dissertation, the derivatizing reagent crystalizes on being

exposed to air blocking the syringe. Hence, the washing is important, without which, the plunger does not return to its original position due to blockages. This causes collisions and breakage of adapter and robot finger in consecutive aspiration steps.

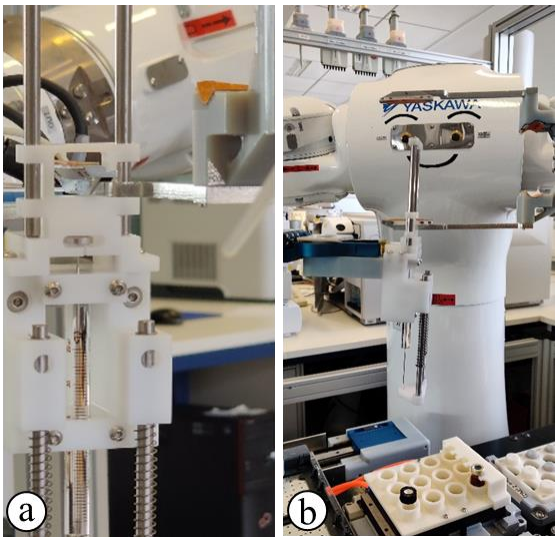


Figure 56: Glass syringe actuated by the robot; (a): fingers of the left arm actuating the plunger, (b): robot arm posture with the cooling rack placed on the ALP

6.5 Summary

The introduction of the SPE extends the range of processes performed by the dual-arm robot in the RS-2. Liquids can be directly pipetted in the SPE cartridges placed on the carriage. The addition of electronic pipettes helps overcome the limitation of using a large number of manual pipettes. A wide range of liquid volumes can be pipetted by just two pipettes. Odd volumes can also be dispensed. This leads to saving in space required for hanging the pipettes. The pipettes do not need to be changed often mid-process thus saving time. The force feedback grippers add an important feature to the robotic system. The robot arm sometimes starts moving to a new position before the gripper action is completed due to communication glitches. Similarly, labware may fall from the grip which could not be detected by grippers in RS-1. The PTM grippers help overcome this drawback and improve the system. Besides they require no extra programming even if new labware is introduced or labware orientations are changed. The gripping force can be changed as required to enhance the grip strength. The addition of glass syringes and a cooling plate also extend the usability and add an important feature to the robotic system. Derivatization processes are now possible with the RS-2. The glass syringe may also be used for other types of liquid transfers if necessary. The cooling plate may be used to keep measurement samples at a lower temperature during the process if required.

7 Basics of Motion Elements

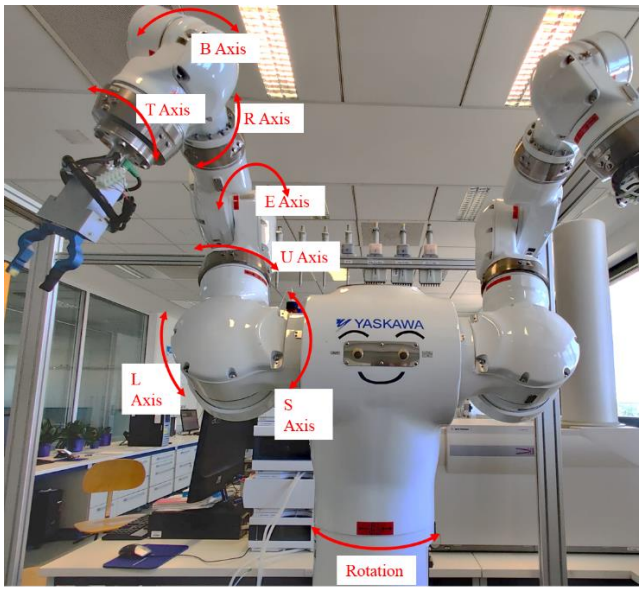
The aim of the dissertation is to program the Yaskawa SDA10F dual-arm robot to perform sample preparation tasks in a time efficient manner. Every sample tested in the lab is different and requires a different set of pre-analysis processes. The system needs to be flexible enough to accommodate future application needs. The sample preparation process involves a number of difficult labware handling motions by the robot arms and programming the dual arms is a difficult task due to the fragile labware and the constricted spaces of the work environment. These motions cannot be created as one long process as it makes the system rigid. Hence, motion elements and task jobs are created using the teach pendant in the Inform III language.

7.1 Robot Posture

The trajectory of motion elements depends upon the arm posture. Each joint of the robot has a fixed range of angular motion and permissible maximum speed (Table 7).

Table 7: Table depicting the rotation limits of each robot arm joint

Robot Joints	Max Motion Range	Max Speed Limit
S-Axis (Lifting)	±180˚	170˚/s
L-Axis (Lower Arm)	±110˚	170˚/s
E-Axis (Elbow)	±170˚	170˚/s
U-Axis (Upper Arm)	±135˚	170˚/s
R-Axis (Upper Arm Twist)	±180˚	200˚/s
B-Axis (Wrist Yaw)	±110˚	200˚/s
T-Axis (Wrist Pitch)	±180˚	400˚/s
Rotation Axis (Waist)	±170˚	130˚/s



The posture of the robot arm and relative angles of the joints influences the compounded reachability of the robot (Figure 57).

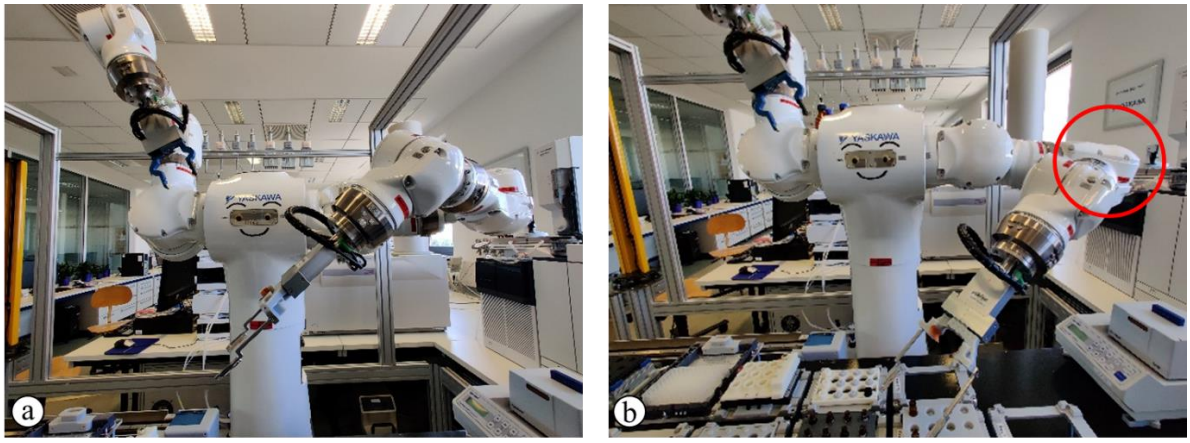


Figure 57: Adjusting the arm posture for maximum reachability, (a) correct position to reach all the ALPs on the workbench, (b) arm posture limits the reachability of the end effector to only a few ALPs

The joint E in Figure 57b has reached its angular limit. The arm can no longer reach the ALPs on the base. Hence, any further motion by the arm is impossible. Alternatively, in Figure 57a, the motion can be carried out easily. Thus, before finalizing the path of the arm, the full range of motion in the particular posture needs to be tested.

Motion elements can be of two types based on their programming method: the Path Motion Element or the Relative Motion Element (Figure 58) [111].

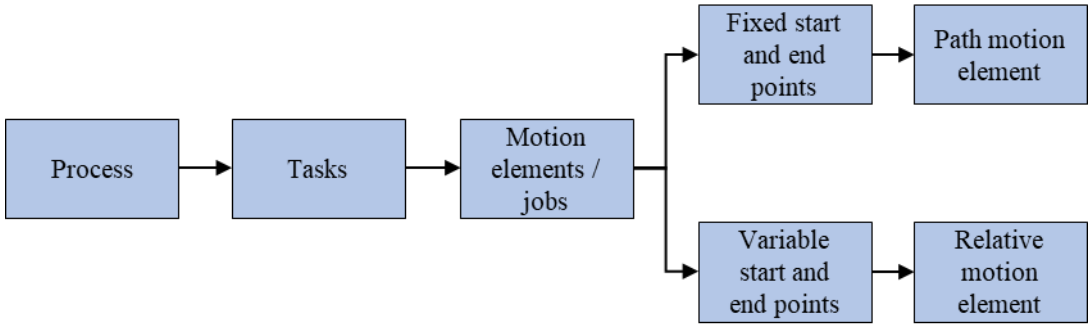


Figure 58: Types of motion elements

7.1.1 Path Motion Element

The path motion elements are programmed by moving the robot using the teach pendant, from the start point to the end point following the desired trajectory. The start and end points and certain points in the path are saved during teaching. The angles of each joint of the arm at each corresponding position are saved in the robot controller (Figure 59). When the job is executed, the robot replays these saved points and follows the said trajectory.

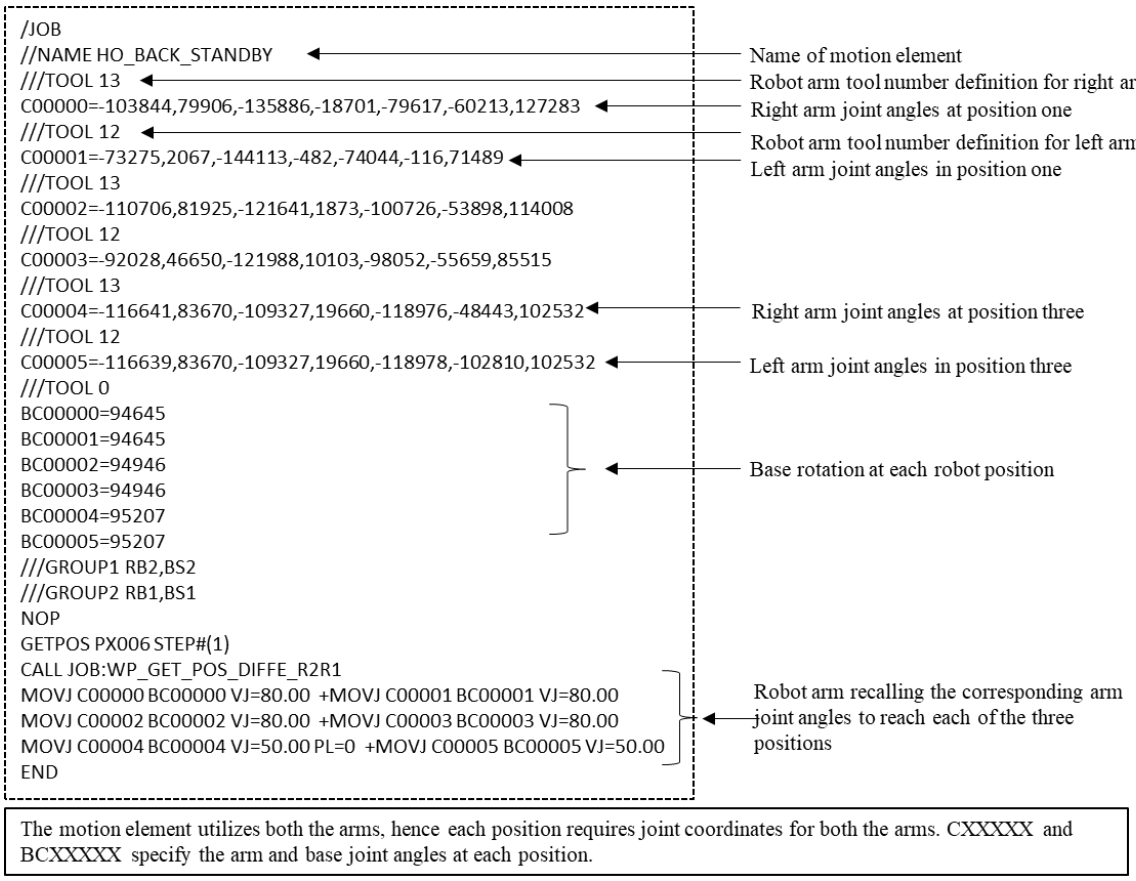


Figure 59: Job content depicting highlighting the robot joint angles for both the arms saved in a path motion

Such programs can be adjusted by individually editing each step. The trajectory of the dual arms is planned such that no collisions between the arms or with any object in the workspace occur. The motion is adjusted precisely around obstacles present in the work area. The robot can reach exact locations in narrow spaces with such programs. However, such jobs are useful only for fixed actions, for example transferring a labware from point A to B. Jobs where the start and end position of the arm is fixed and will not differ in the future are made in such a method.

7.1.2 Relative Motion Element

The robot is required to access a number different positions and labware placed on the working deck during the sample preparation process. The task to be performed is common, e.g., picking up a vial, but the position is different in each case. In such scenarios path element motions are often impractical, as identical motions for each possible position have to be created. The motions created for such cases are called relative motion elements, where the motion of arm is relative to a reference point in a specified coordinate system. The basic motion of the arm to and from the reference point is taught. Distances in the X, Y, and Z directions are added or subtracted using arithmetic calculations, in order to move the arm to the relative points (Figure 60).

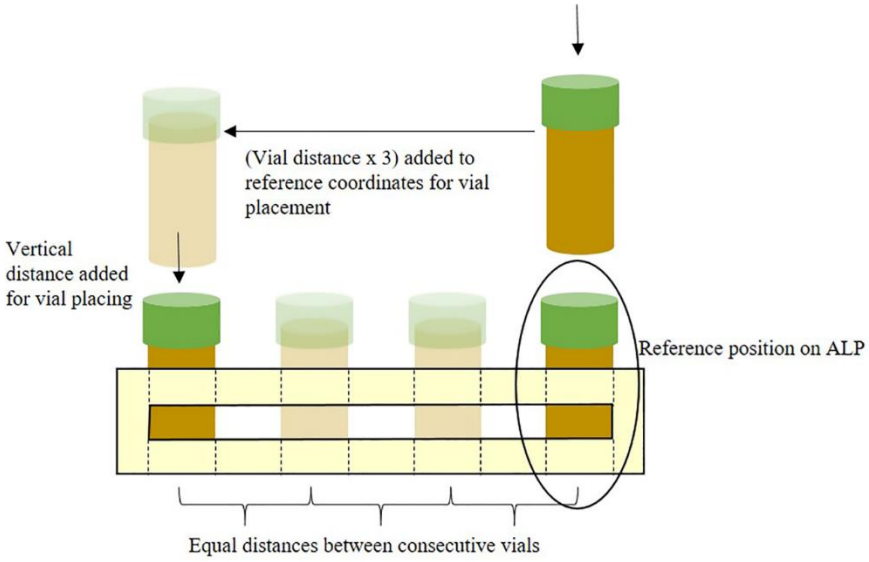


Figure 60: Graphic depicting the algorithm for a relative motion element

In the above example, the figure depicts the motion of placing a vial in a rack. The robot needs the following information to successfully carryout the process: type of vial, ALP number on which the vial rack is placed and the vial number in the rack.

The vertical and horizontal distance to be travelled by the robot arm depends upon the size of the vial, which is dictated by the vial type and the vial number. Vials smaller than 40 mL volume are arranged in a 4x3 configuration, whereas 40 mL vials are arranged in a 3x2 configuration. The reference position of rack is connected to the ALP number. This information is set via the R-Interface 2.0 before execution of the job, to define the exact position where the arm is required to move.

7.2 Reference Points for Motion Elements

Different applications require differing labware and workbench configuration. Hence, it is necessary that the robot is capable of reaching any ALP and performing any task. It is not practical to create individual robot motions for all positions. Hence, the jobs were designed on the principle of ‘motion frames. Motion frames is the mapping of a motion relative to various reference points. The reference points for motion frames affect the flexibility and accuracy of the system. These points are determined before programming a motion element and should be selected based on the labware handled and tasks carried out. The posture for reference points has to be chosen such that all the necessary positions are reachable.

The tip boxes from the shelves and labware from hotels are picked up by the left arm. Thus, a motion frame for both the storage options is defined with the reference points of the left arm. The basic motion to reach the first position of a shelf and the hotels is taught. The robot arm requires to travel horizontally and vertically relative to the reference point to reach the various hotel and shelf positions. The labware 0 in hotel 0 and shelf 0 is considered as the points of reference and all calculations are done accordingly (Figure 61).

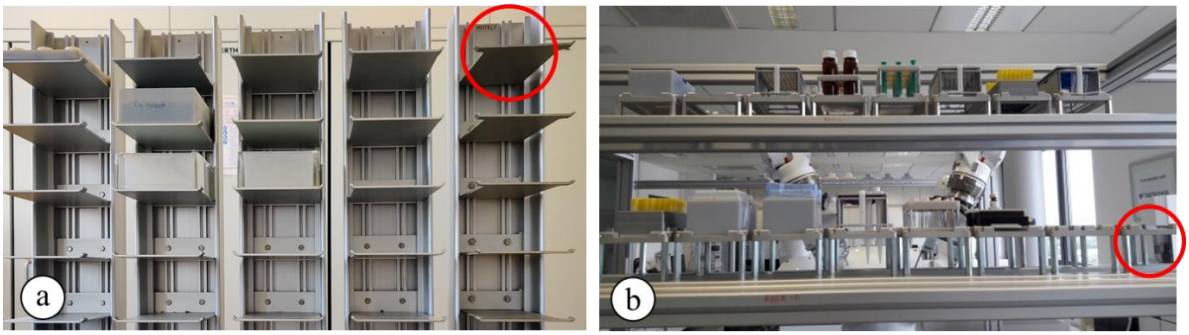


Figure 61: Reference points (a) Hotel 0 room 0 as hotel reference point, (b) shelf position 0 as shelf reference point

All the labware is brought down to the workbench during actual working and a number of complicated tasks with both the arms are performed here. A task should be performed using labware placed on any of the ALPs. Thus, the robot motions are mapped by defining reference positions for all the ALPs. The RS-1 implemented a single reference point system, i.e., the fingers of the right arm were made to touch the upper right corner of each ALP and the posture coordinates of the arm were used as reference coordinate for the particular ALP (Figure 62).

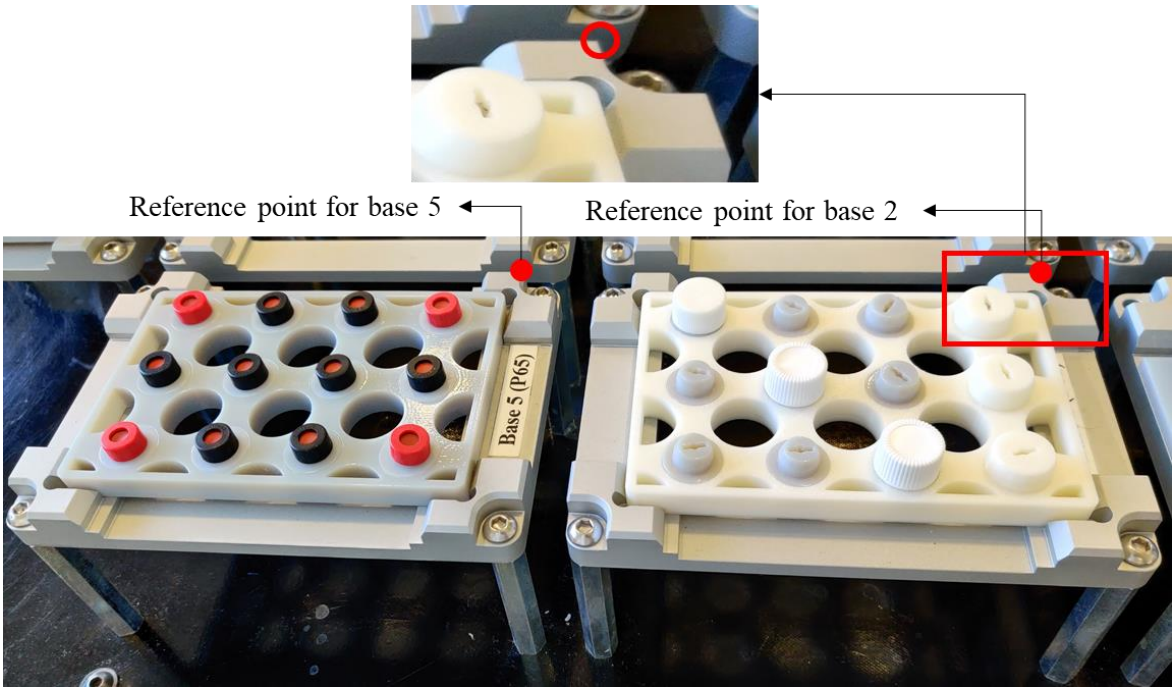


Figure 62: Reference point for each ALP according to the existing system [111]

However, it is impossible to select identical points on all the ALPs. Thus, motion elements for picking vials, syringes etc. had to be customized for each ALP using 'if condition' blocks. For example: if ALP = 1, move arm 300 mm in X-direction, if ALP = 0, move arm 200 mm in X-direction. In case of motions with the left arm, like picking or placing labware on the ALPs, the reference positions were translated to the right arm and used for further calculations. Thus, minor positional changes in motion elements involving left arm required a recalculation of the coordinate translation between left and right arms. Besides any changes in reference positions of the right arm affected all motion elements.

In order to allow independent editing of single arm motions elements and to simplify the creation of motion frames, the RS-2 implemented independent reference points for left and right arms (Figure 63). Both the arms mostly perform distinct independent tasks. Having distinct references eliminates the need to import coordinates between the two arms and additional calculation steps required. Furthermore, the reference points of one arm can be adjusted at a later date to modify the jobs without affecting jobs done by the other arm.

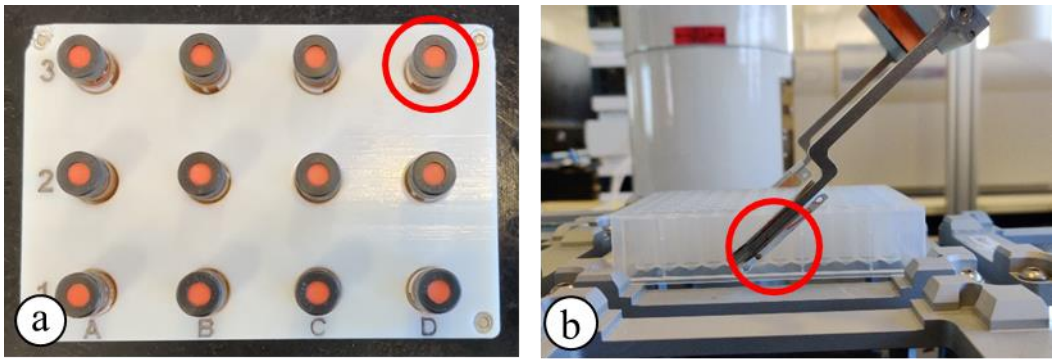


Figure 63: Reference points on ALPs for (a) right arm, (b) left arm

The right arm requires to pick vials, vial lids, syringes from the workbench. As seen in Figure 63a, the position of the top right vial in a 2 mL vial rack is considered as the reference point for the right arm. As 4x3 is the most commonly used configuration, this labware is selected for the reference. For 40 mL vials set in a 3x2 configuration, a fixed distance is added to the x and y directions at the start of the motion in order to align the fingers with the first vial. Further, the resultant coordinates are operated upon normally to carry out the labware manipulation. The right arm is positioned vertically above the vial. This ensures that the reference point on all ALPs is identical with respect to the labware. For creating relative motion elements, the reference point can be considered as the 1st position and subsequent vial positions can be calculated with primary arithmetic operations which are common for all the ALPS reducing programming effort of the operator. The reference position of one ALP can be changed without affecting the others.

Similarly, as seen in Figure 63b, the reference point for the left arm is the gripper position for picking/ placing the labware on the rack. The reference position allows to exactly adjust the labware to the ALP. Error in labware placement on any of the ALPs can be fixed by editing the reference point. Motion elements carried out by the left arm are independent of the right arm and hence changes in either of the reference points will not affect the other. All similar labware are held at the exact same position bringing uniformity to the program.

8 Generation of Motion Elements

The improved robotic system (RS-2) has been programmed to carry out fifteen different tasks namely: pipette task (PE), electronic pipette task (EP), glass syringe task (GS), glass pipette task (GP), syringe task (SY), vial tasks (VI), workbench tasks (TA), hotel transfer task (HO), shelf transfer tasks (TI), shaker task (SH), ultrasonic bath task (US), LC task (LC), GC task (GC), thermoshaker task (TM) and SPE task (SPE). At the beginning of each process the robot arms move from the standby position to the intermediate position followed by the task sequence for sample preparation.

8.1 Standby Position and Intermediate Positions

Figure 64: Standby Position

The standby position (Figure 64) is the neutral position with the robot arms folded close to the robot body. The robot is stowed in this position when not in use. The arms of the robot do not interfere with themselves or the robot environment in this position. In the original control software (RS-1), all robot tasks start in the standby position. The R-Interface requires the robot arms to return to this position to signify the end of a task so that the next command can be accepted by the robot controller. However, the motion of bringing the robot arm back to this position after each task is time consuming. Each task is carried out independently and there is no smooth combination of different tasks. In addition, it looks repetitive to the observer.

Figure 65: Intermediate Position

In order to reduce the time spent in positioning robot arms to start performing a task, a new robot posture was required in RS-2 such that the motions were shortened compared to RS-1 or could be completely eliminated. Hence, a new position called *intermediate position* is introduced (Figure 65). This position is closer to the labware than the standby position. As can be seen in the images, the posture of both the arms is different as both arms have different functionalities. The left arm is used to transfer labware from the storage areas. It requires a flexible posture such that the workbench, hotel, shelf and the SPE is reached with minimum effort. The left arm is already in position for manual pipette jobs that require the pipette piston to be pressed. The right arm is used to pick up vials, or instrument lids and travels mostly in the vertical direction. Thus, the right arm too does not need an additional motion element for posture adjustment saving time. It also reduces the total number of motion elements required

and the programming effort therein. The robot is moved by the R-interface 2.0 from the standby position into the intermediate position at the beginning of a process. The robot is then ready to accept commands for performing the various tasks. The robot need not travel to the intermediate position in between tasks to signify the end of a task. At the end of the sample preparation process, the R-Interface 2.0 moves the robot arms back into the standby position during inactivity

8.2 Node Points and Key Points

Each motion element has two node points: a start node and an ending node. A node is the arm posture created by a specific combination of arm joint angles. The start and end nodes of a motion element were always non-identical in RS-1 so that a motion could not be carried out repeatedly in a loop. This was done in order to stop the robot from carrying out wrong motions while holding a labware. Although this approach added an element of safety to the system, it made the system less flexible. The node points are programmed in such a way, that the motion elements can be connected in only one way. No other combinations of existing motion elements can be created to form a new task if required in the future. In such an event, new motion elements will have to be created by the user which is time consuming. In addition, the user requires special training for creating the motion elements and knowledge of how the existing node points are connected, as well as the impact of changing a node point. Thus, the flexibility of the system in terms of application by the user is reduced.

In order to expand the usability of the system, node positions have been altered in RS-2 which required re-planning and reprogramming of the motion elements. The start and end nodes of a motion element may be identical or different depending upon the action performed by the robot. The node points are important to a particular task. They help in maintaining continuity between consecutive motion elements of a task. Node points can be of three types: fixed, floating or mixed (Figure 66).

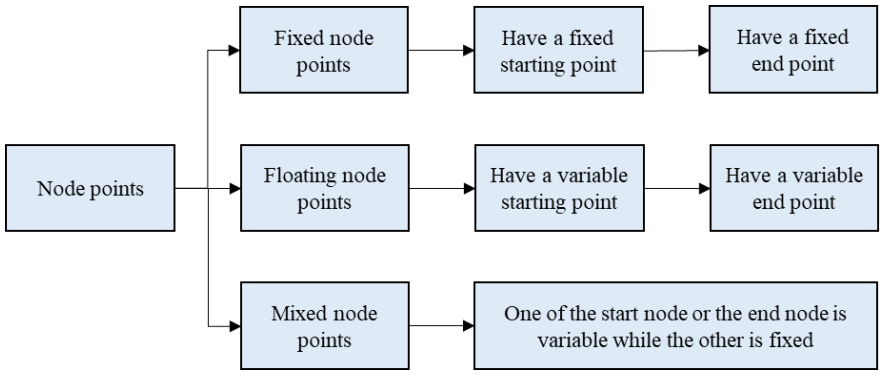


Figure 66: Node classification

Fixed node points are introduced where the motion elements require a defined start and end position. Fixed nodes were also seen in RS-1. However, the start node and end node of a motion element in RS-1 was not allowed to be identical. The motion elements were programmed so on purpose so that a single motion element could not be carried out in a loop and would trigger an error message if attempted. As seen in Figure 67a and Figure 67b two different motion elements are required for picking and placing vials from an adaptor. In RS-2 however, identical node points are given importance as they allow motion elements to be connected to each other

(Figure 67c). Besides, they are valuable for error handling as the robot can retrace the motion backwards if required.

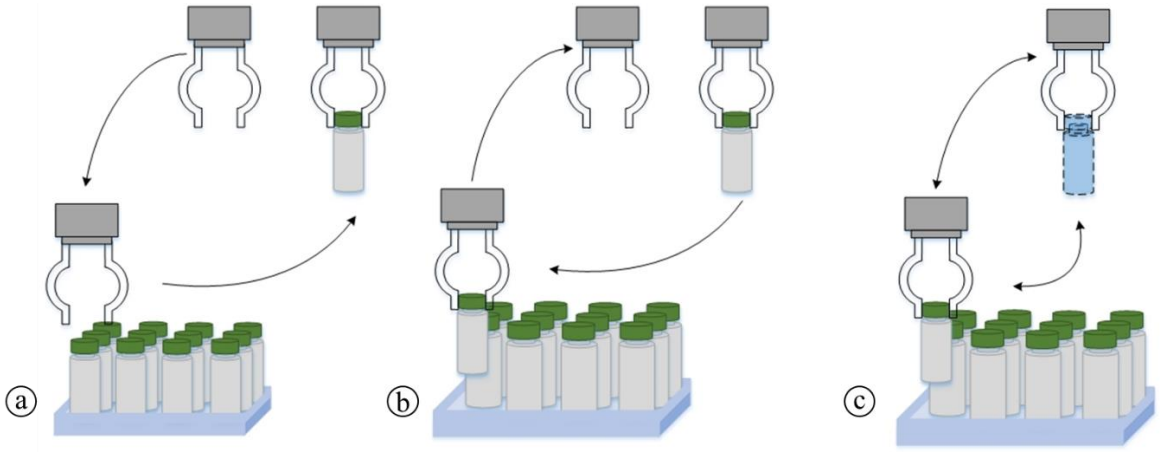


Figure 67: (a) pick vial motion element with different start and node points in RS-1, (b) put vial motion element with different start and end node points in RS-1, (c): single motion element with identical start and end node points for picking and placing vials in RS-2

In order to add smoothness to the robot motions and eliminate unnecessary movement, some motion elements do not have a specific start or end point. Such floating end points reduce arm travel and shorten the time between consecutive motions. For example, the arm need not travel to a fixed position after aspirating liquid from an ALP, instead can directly move to a new ALP for dispensing. The position of the arm after each aspiration and dispensing might be different depending upon the ALP being used. Some motion elements may have mixed node, that is one fixed node and one floating node. The motion element for rejecting pipette tips is an example of such a motion element. The arm can be at any position after dispensing liquid on one of the ALPs, the motion element to release tips starts at this floating node and then follows a fixed path taught by the programmer to end at a fixed position so that the pipette is ready to be put back to the shelf.

The biggest drawback in freely transitioning from one task to the other is the extreme difference in robot arm postures for the varied tasks. For example, the robot arm is in front of the robot at the end of a vial opening task, and needs to travel to the back of the robot environment in order to pick up a manual pipette. In case of consecutive vial opening and pipetting tasks, the robot cannot change its posture independently due to limitation in join rotation. Hence the robot tasks always started and ended at the standby position in RS-1 to overcome the obstacle. In order to allow the robot to transitions between various tasks without having to go back to the standby or intermediate position often, four different arm postures (nodes) have been identified as ‘Key Points’ in RS-2 as seen in Figure 68. Key points assist the robot controller in identifying its relative position in the workspace and define the start and end positions of the robot arm in a task. The key points play an important role for seamless transitioning between various tasks and re-integrated in the motion planning algorithm which is explained in a later chapter.

Figure 68: (a) Hotel key point, (b) workbench key point, (c) shelf key point, (d) pipette key point

Four motion elements called 'preparation' motion element are programmed to start from the intermediate position. This motion element changes the arm posture to one of the key points such that it is suitable to carrying out the planned task. The postures at key points are shared by multiple tasks. Thus, any task ending at one of the key points can be followed by another task starting at the same key point without having to move the robot arm back to the standby or intermediate positions.

Table 8: Key points and corresponding tasks that can be carried out without moving the robot to the intermediate position

Key Point	Tasks connected to the key point
Hotel key position	Hotel task SPE task Workbench task
Shelf key position	Shelf task SPE task Workbench task
Workbench	Workbench task Shelf task Hotel task Thermoshaker task
Pipette	Pipette tasks

The vial task, electronic pipette task, thermoshaker opening, LC opening, GC task, shaker task and ultrasonic task start directly from the intermediate position and do not need a 'preparation' motion element. Changing the arm posture directly from one key point to another is not possible due to rotational limitations of the arm joints. Hence, for consecutive tasks ending and starting at different key points, the arm has to be put in the intermediate position. However, unlike RS-1, this need not be done after every task. The key point arm postures cater

to all areas of the robot environment and are hence, important from a future expansion point of view. Key points let the programmer connect various tasks while retaining the independent structure of each task. New tasks added in the future can be joined to existing tasks via the key points making consecutive tasks seamless and transitioning less time consuming making the overall process faster than in RS-1. The five key points cater to the entire working area. In the event that a new task needs to be added to the automated process, if the labware is placed within the working area, one of the five key positions would be already compatible to the new task. Thus, the programmer only needs to create a task module in the interface program, specify the sequence of motion elements and connect it to the relevant key point.

The reduced number of node points and key points are introduced as a safety feature and add uniformity in all the motion elements. The key points help in categorizing the motion elements required for the various groups of tasks. In the future, any new motion element to be created can use these key points as the reference. Any new motion element starting and ending at one of these points can automatically be used in continuation with other motion elements made for that particular task. Thus, adding new motions for new processes is made easier. The key points tip shelf, hotel and workbench are serviced by the left arm. The pipette position, is serviced by the right arm. The intermediate position is used by both the arms.

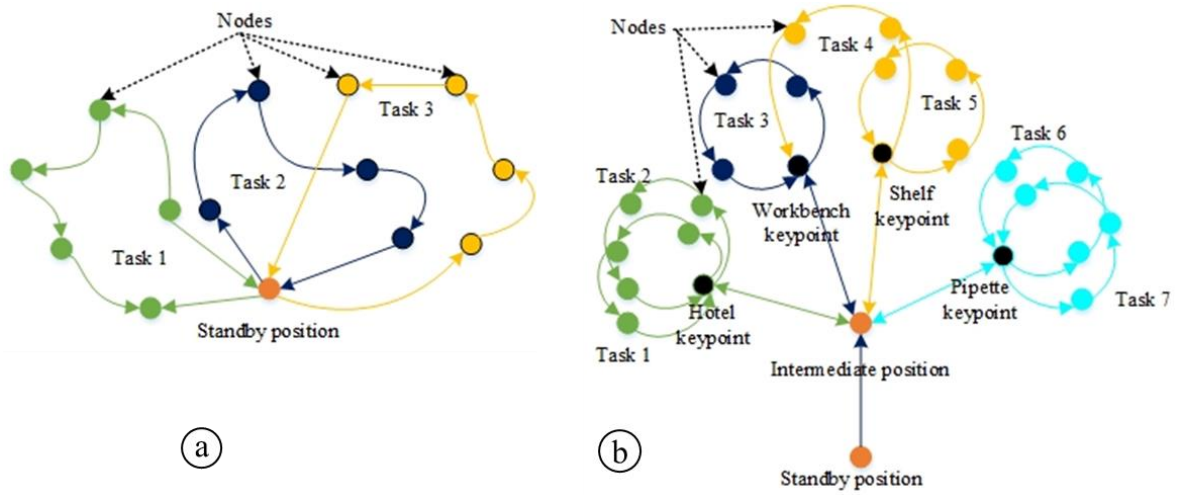
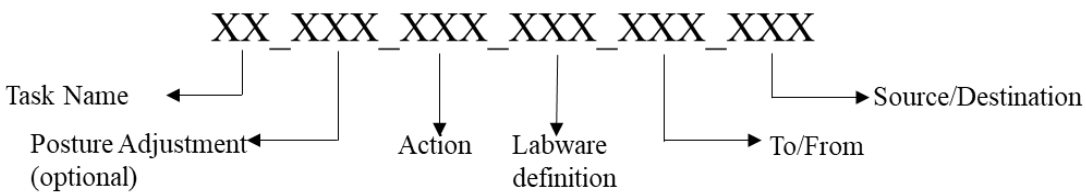


Figure 69: Task design, (a): All tasks starting and ending at the standby position in RS-1, (b): Tasks starting and ending at one of the key points for continuous and flexible task planning in RS-2:

8.3 Naming of a Motion Element

Each motion element in the RS-2 is named according to the actions performed during the motion and also include the task module name, labware and destination. For example:



The names can be alphanumeric. This type of naming system makes it easy to identify the purpose of the motion element from the name.

Which task the motion element belongs to, is determined by the resting position of the labware. For example, when the labware is picked from the shelf, the name of the motion element starts with 'TI' (corresponding to 'tip shelf'). However, the motion element that places the labware on the workbench starts with 'TA' (corresponding to 'table'). These two motion elements are run consecutively to transfer the labware from the tip shelf to the workbench, but have different purposes, are linked with different key points and hence have different task names. The inclusion of task name is important for the job search algorithm which is explained in detail in later chapters. The four key points are connected to the interface position by means of special 'prepare' motion element and can be identified by the 'PRE' included in the motion element name. Such motion elements indicate that the robot is moved from the intermediate position to the key point. The grasping action is not done in this motion. An example of such a motion element for moving the robot arm from intermediate position to the hotel key point is-

HO_PRE_PICK_LW_F_HOTEL.JBI

HO represents that a hotel transportation task is carried out. Hotels and shelves are used to store plate type labware such as 2 mL vials, MTPs, reservoirs, lid racks etc. The labware size is standard. Multiple labware make use of common motion elements which is indicated by the use of 'LW' in the motion name. The motion element also indicates the action being performed ('PICK', 'PUT', 'OPEN', 'CLOSE'). A motion element to pick plate type labware from the hotel is named as-

HO_PICK_LW_F_HOTEL.JBI

However, certain labware have a designated position on the workbench and require a unique transportation motion. The cannula rack and filter rack for syringes are examples of such labware. These labware are transported directly from the hotel to the destination position without changing the grip angle and require a special motion. Such unique motions include the name of the labware that they are created for.

HO_PUT_CANNULA_T_TABLE.JBI

Lastly, it is also necessary to define the direction of motion through the name of the motion element for the user to be able to distinguish between the various motions. The direction of motion is indicated by the last two terms in the name of the motion element. The term 'T' or 'F' stands for "to" and "from". The motion element where a vial lid is placed onto the lid rack is named as-

VI_PUT_LID_T_RACK.JBI

The above name indicates, that a labware (already grasped and picked by the robot) is placed on a lid rack placed on any of the ALPs on the workbench during this motion. All motion elements have a .JBI extension.

8.4 Designing a Motion Element

For the robot to carry out a task, all the relevant motion elements have to be present on the FS100 controller during robot working. The FS100 has the capacity to store 10,000 motion commands in total at a given time. Motion commands are the commands within a motion element, where the robot arm is instructed to move. The RS-1 had a large number of motion elements with short arm motions that exceeded this limit. (It is not possible to count the exact number of motion commands in total as that would have to be done manually by reading each and every motion element individually). Thus, all the motion elements could not be loaded to the controller at a time, restricting the number of tasks performed during a process. This limitation hinders the performance of RS-1. This is a major obstacle when carrying out complicated sample preparation processes involving multiple tasks.

The robotic system can be said to majorly undertake transportation of "objects", "objects" being either labware in the form of MTPs, vials, syringes, pipettes, etc. or liquids in terms of liquid transfers. Hence, in the general sense, it can be understood that something is picked from one place and put in the other. In RS-1, the programmer has divided all the tasks in five major parts: prepare to pick object, pick object, prepare to put object, put object, return to original position (Figure 70).

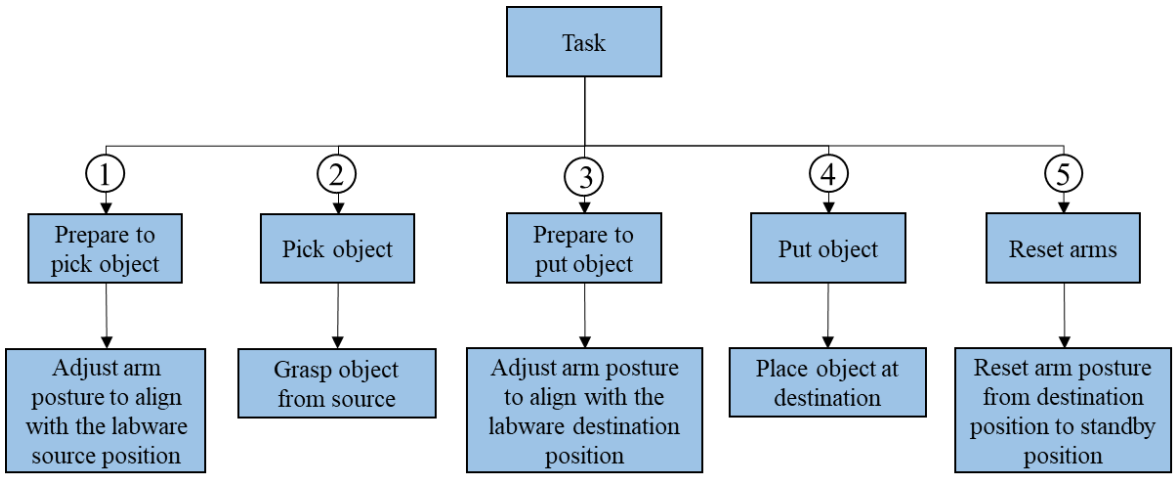


Figure 70: Subdivision of a task into five primary motion components

That is, one transportation task is divided into minimum five or more different motion elements. The 'prepare' motion elements do not serve a purpose apart from changing the robot posture. Labware transportation motions in RS-1 have a limited range. All tasks could not be performed on all the ALP positions available on the workbench. Hence individual motion elements have been made for individual ALPs. A similar trend is seen for jobs created to pick labware from hotels or shelves (Figure 71). The wide range of labware requires varied grip positions and thus individual transportation motion elements are seen for the different labware being transported via the same path. That means for any new labware introduced in the robot environment, new motion elements have to be created. This lends too much fragmentation and possibilities for positional errors or continuation errors. Such type of motion programming made the motion element database in RS-1 bulky with 274 motion elements [111], which in

turn makes the interface programming and SAMI Ex usage complicated. Long sequences of motion elements have to be called while operating the robot on teach mode.

An example of the list of various motion elements for transportation of different labware from the hotel to the workbench is given in Figure 71.

HO_AFT_PUT_MTP_T_HOTEL	Back to standby after placing microplate (MTP) in hotel
HO_AFT_PUT_MTP_T_TABLE	Back to standby after placing microplate (MTP) on table
HO_PRE_PUT_MTP_T_HOTEL	Prepare to put microplate (MTP) to hotel
HO_PRE_PUT_RES_T_HOTEL	Prepare to put reservoir to hotel
HO_PRE_PUT_RES_T_TABLE	Prepare to put reservoir to table
HO_PICK_MTP_F_HOTEL	Pick microplate (MTP) from hotel
HO_PICK_RACK_F_HOTEL	Pick vial rack from hotel
HO_PICK_MTP_F_TABLE	Pick microplate (MTP) from table
HO_PICK_RES_F_HOTEL	Pick reservoir from hotel
HO_PICK_RES_LID_F_TABLE	Pick reservoir lid from table
HO_PICK_CANNULA_F_TABLE	Pick cannula from table
HO_PUT_CANNULA_T_TABLE	Put cannula to table
HO_PICK_FILTER_F_TABLE	Pick filter from table
HO_PUT_FILTER_T_TABLE	Put filter to table

Figure 71: List of some motion elements for transferring labware in and out of the hotels as seen in RS-1

The motion element search process in manual mode is time consuming if too many changes are required to take the robot from one posture to another. It can prove to be complicated and time consuming for a new user leading to collisions and error messages. The result is a complicated system which goes against our goal of a functional and simplistic automation solution. Hence, it was necessary to reduce the total number and size of the motion element files so that they can all be contained on the controller. In addition, with the introduction of new nodes and key points, the motion elements for the tasks had to be re-planned.

The robotic system would be used to carry out various processes, involving a combination of the tasks and a variety of labware. It is expected to be able to carry out motions with varying repeatability and in varying sequences in order to perform new processes in the future. Hence the robot motions should be planned and programmed to provide flexibility of process design. An optimal robotic system would be one which required the least amounts of changes or new additions to be done when any new process is introduced. These include addition of new motion elements, usage of system variables, usage of storage spaces, changes in labware and labware placement during processing, etc. Thus, the previously existing motion elements were re-planned keeping in mind all the constraints. Additionally, the replacement of the robot grippers also made it necessary to change the motion elements in order to suit the new gripper orientation in RS-2.

The fragmentation of tasks into motion elements has to be planned in a way that it does not make the system too rigid for future editing. Fragmenting tasks into too many motion elements would make the system flexible as they can be combined in more ways to create new tasks. However, they would make the system unnecessarily complex and difficult to maintain in terms

of motion continuity. Additionally, programming of a motion element also needs to be done keeping in mind the possible future system expansions.

The introduction of key points and common 'prepare' motion elements for multiple tasks has minimized the number of motions that do not involve labware handling and are required purely for arm posture adjustment (Table 9).

Table 9: Table representing in short, the changes to planning of motion elements

Motion Elements in RS-1		Motion Elements in RS-2	
VI_PICK_VIAL_N-LID	Pick vials without lid	VI_SJ_PICK_VIAL	Pick vials (with or without lids)
VI_PICK_VIAL_Y-LID	Pick vials with lid		
VI_PUT_VIAL_N-LID	Put vials without lids	VI_SJ_PUT_VIAL	Put vials (with or without lids)
VI_PUT_VIAL_Y-LID	Put vials with lids		
VI_PRE_PUT_LID_T_RACK	Prepare to put lid to rack	Equivalent motion elements are not required in RS-2 as the arm is in prepare position by default.	
VI_PRE_PUT_VIAL_T_RACK	Prepare to put vial to rack		
VI_PRE_PICK_LID_F_RACK	Prepare to pick lid from rack		
VI_PRE_PICK_VIAL_F_RAC K	Prepare to pick vial from rack		

The pipetting task is the most extensively used and requires the highest number of motion elements. Two types of manual pipettes are available: single channel and multi-channel. There are 9 pipettes in total. The pipettes tips need to be loaded to each pipette. They transfer liquid from either vials or MTPs or reservoirs. The liquid is pipetted to either vials or MTPs. The labware can be placed in any of the 13 ALPs. In addition, there are five different types of vials. Each pipette, labware combination has a unique motion element in RS-1. Within the motion element, there is separate calculation for reaching the different ALPs. Thus, for the manual pipetting task alone there are 76 motion elements.

Hence, motions for the manual pipette task were re-classified and re-fragmented to create new motion elements that are concise. For example, the three single arm motion elements for preparing arm to load tips, load tips and preparing arm to aspirate liquid have been shortened into one single motion elements with both the arms used synchronously. Thus, a number of motion elements for pipetting task can be easily eliminated as can be seen in Table 10.

Table 10: List of motion elements for a manual pipette task required in RS-1 and RS-2

Motion elements in RS-1	Motion elements in RS-2
Prepare to pick pipette	Prepare to pick pipette
Pick pipette	Pick pipette
Prepare to load tips	(not required)
Load tips	Load tips
Prepare to get liquid from vial	(not required)

Get liquid from labware	Get liquid from labware
Dispense liquid to labware	Dispense liquid to labware
Release tip from pipette	Release tip from pipette
Prepare to put pipette to rack	(not required)
Put pipette to rack	Put pipette to rack
Back to standby position	Back to intermediate position

In RS-2, the total number of pipetting motion elements has been reduced to 45. These motion elements cater to all the existing labware and pipette volumes and the various ALP positions. Only one motion element is created for aspirating liquid and is common for all single channel pipettes. Similarly, one motion element is created for liquid aspiration using all multi-channel pipettes. The basic code in a liquid aspiration motion element is common for all vials, the difference is only in the pipette depth depending upon the volume of the container. If a new vial size is introduced, the existing motion element for aspirating and dispensing liquid will only need the addition of a few code lines to adjust the pipette depth (Figure 72). The same programming method is followed for lading tips, releasing tips, dispensing liquid as well.

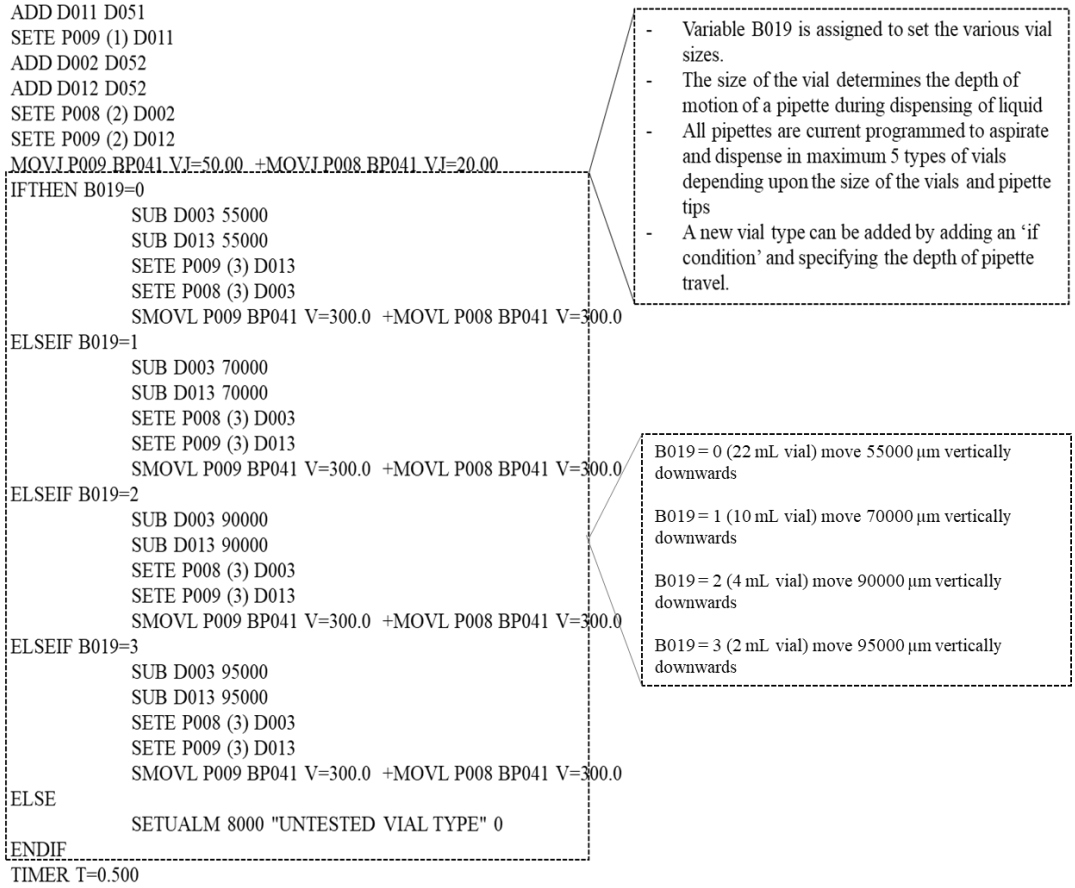


Figure 72: Graphic showing an example of a motion element for aspirating liquid from multiple labware cases using a manual pipette

Similarly, the hotels store labware like the MTPs and reservoirs, etc. The motion elements have been combined such that the common robot motion follows a common path to reach the labware. The labware to be picked up uses a condition in the motion element. The set condition dictates the change in certain parts of the motion element code. The change might be grip

position, gripping force, presence of a lid etc. Various variables that have to be adjusted before a task is carried out. For flexible motion planning it is important that the number of variables used are minimized and optimized. Such fragmentation allows the programmer to include new types of labware to the existing program for future applications. The database also does not become bulky and unmanageable. Thus, motion elements from RS-1 have been re-designed in order to reduce the number of files, bringing the total number of motion elements down to 154 in RS-2. All the motion elements can now be loaded onto the FS-100 controller. Thus, all programmed tasks are available to be run on the robot for a given process thus overcoming the drawback of the RS-1.

9 Flexible Motion Planning

9.1 Combining Motion Elements

The motion elements created using the teach pendant are saved in the FS100 and also on the controlling PC. The files from the PC are imported by the R-Interface into the motion element database at the beginning of the sample preparation process. The file names have a .JBI extension. A job search algorithm is written by the programmer to find the correct motion elements from the database while the motion combination module sets the motion elements in the desired sequence and writes it to a file called a 'task job'. A task job is a .JBI file through which the motion elements are called in the correct sequence. The task job is created by the motion combination module of the R-Interface and sent to the FS100 controller for execution. In addition to the task job, all motion elements listed in the task job need to be present on the FS100 controller for the robot to be able to execute them (Figure 73).

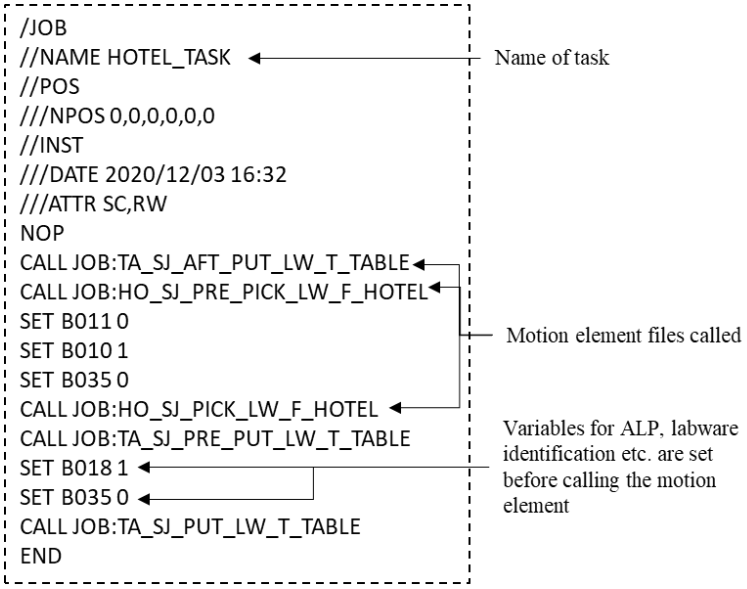


Figure 73: Example of a hotel task job calling the various hotel motion elements in sequence and setting the required variables

The job search algorithm in RS-1 was long and confusing to a new programmer. Hence from a future development perspective, it needed to be streamlined. In addition, the introduction of redesigning of motion elements in RS-2 also required the motion combination algorithm to be changed in R-Interface 2.0. The various differences in R-Interface and R-Interface 2.0 and the effect of the changes on RS-2 have been outlined in the following sections.

9.1.1 Job Search Algorithm and Directory Refresh

The node points help the R-Interface to identify consecutive motion elements. Hence, if any changes are made to the node points, the motion element file becomes unrecognizable to the R-Interface. In addition, all the jobs are programmed with different starting points in order to make it easier for the interface program to distinguish between the jobs. This creates a number of discontinuities in robot motion planning and a large number of node points for the programmer to remember and keep track of.

Each task performed by the robot is coded as individual task modules in the R-Interface in RS-1. A sequence of motion elements to fulfil the task is specified by the programmer for each task module in proper word form, for e.g.: “Prepare to Pick Microplate from Hotel”. At the beginning of each task, a temporary copy of the entire motion element database as an Array List is created by the R-Interface in the local memory. The file names of the motion elements have a .JBI extension, for example: “HO_PRE_PICK_MTP_F_HO.JBI”. Each motion element has numerous attributes such as the node points, robot arm used, compatible labware types, compatible positions in the robot environment, etc. All these attributes are connected to the name of the motion element and are saved as an Array. The SAMI Ex sends commands for the first task to the R-Interface and the relevant task module is triggered. The job search algorithm then accesses the motion element database copied at the beginning of the task and reads the name of each motion element serially.

In order to match the motion element names to those specified by the programmer in the task module, the programmer created a list (word list) of all the words used to define the names of motion elements. These words were mapped to all the terms used in names of the motion elements in the .JBI format (Figure 74). The name of each motion element is thus translated and the specified motion element is found and selected.

Terms = Words	
PRE= Prepare PICK= Pick MTP= Microplate RES= Reservoir F= From FR= From T= To TO = To TA= Table TABLE= Table … … …	… … **PRE_PICK_MTP_F_HO = Prepare to Pick Microplate From Hotel** **PRE_GET_L_F_VI= Prepare to Get Liquid From Vial** … … …

Figure 74: Table depicting keywords and motion element translations included in old system design

To select the next motion, the robot interface first reads the end node of the previously selected motion element. It then accesses each motion element in the database to read the start node of each file. All motion elements with matching start nodes are shortlisted. The names of each file in this list are translated and searched serially to find the appropriate file based on a number of conditions such as file name, labware used, shelf positions, pipette volumes etc. these conditions differ for each task. Often a specific key word from the file name was used to make the search easy (Figure 75).

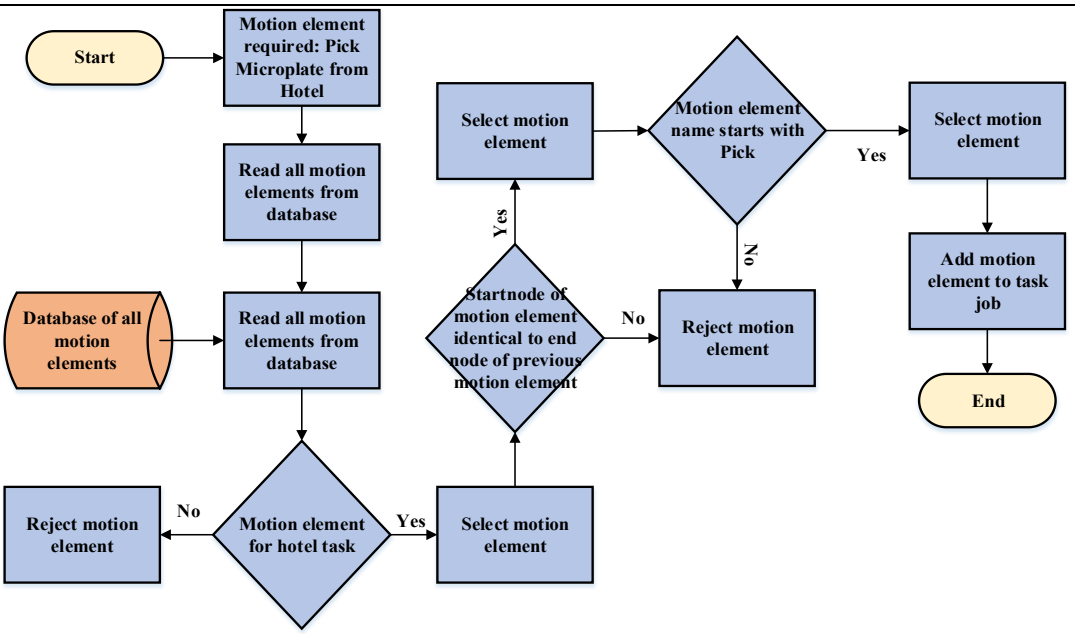


Figure 75: Algorithm for finding consecutive motion elements in a task according to RS-1

The job search worked well without errors except when two motion elements having the same starting node points and containing the same key word in their names existed. The program then selected the first instance available without confirming the entire name of the file. For example, if 'Pick Reservoir From Hotel' is listed before 'Pick Microplate From Hotel', the first instance will be selected. This resulted in wrong tasks being picked up by the system and potential for collisions and errors during runtime. All the required motion elements were thus selected and compiled into one task job. The motion elements were not detected if the node points did not match either, resulting in error messages.

The word list had to be updated when a new motion element or new term was introduced. Every new abbreviation used in the motion element names had to be added manually into the list. Spelling errors and missing terms lead to errors in job search. It was observed, that some words were mapped to multiple terms as the list was long and unmanageable. These translation of the motion element names did not serve a specific function apart from searching the motion elements from the database. Besides, each task contains a list of minimum five motion elements, for example consider this motion sequence for open vial task

-VI_SJ_PICK_VIAL.JBI
-VI_SJ_TRAN_VIAL_T_R1.JBI
-VI_SJ_OPEN_LID_T_VIAL.JBI
-VI_SJ_TRAN_VIAL_T_R2.JBI
-VI_SJ_PUT_LID_T_RACK. JBI

If all 12 vials on an ALP are required to be opened, the above sequence is iterated 60 times (5 motion elements x 12 vials). Each time the database is recalled and the job search algorithm runs through the entire database serially to reach the right motion element to be used. It requires time as well as computational effort as the job database comprises of 274 files with multiple attributes.

In order to overcome all the previous obstacles, the job search algorithm in R-Interface 2.0 is enhanced with the use of 'Directory'. A directory is a table of motion elements mapped to a code name which helps shorten the job search. The code name is a unique value and multiple motion elements can be mapped to one code name. The word list has been completely eliminated. Thus, names of the motion elements do not have to be translated from the .JBI format to normal word format as in RS-1. Instead, the names of motion elements in the task module are specified in the .JBI format by the programmer. As seen in the previous section, all motion element names start with the task name, which is used as a 'code name' for mapping related motion elements in the directory. The dictionary contains fifteen different code names for the fifteen tasks that the robot is capable of performing. Each code name is mapped to the motion elements pertaining to the particular task. An example is shown in Table 11.

Table 11: Example of directory with motion elements listed against the corresponding code name

Code name	Motion elements mapped
VI	VI_SJ_PICK_VIAL.JBI
	VI_SJ_TRAN_VIAL_T_R1.JBI
	VI_SJ_OPEN_LID_T_VIAL.JBI
	VI_SJ_TRAN_VIAL_T_R2.JBI
	VI_SJ_PUT_LID_T_RACK.JBI
HO	HO_SJ_PICK_CANNULA_F_TABLE.JBI
	HO_SJ_PICK_FILTER_F_TABLE.JBI
	HO_SJ_PICK_LW_F_HOTEL.JBI.JBI
	HO_SJ_PRE_PICK_LW_F_HOTEL.JBI
SY	SY_SJ_GET_LIQUID_CANULA.JBI
	SY_SJ_OUT_LIQUID_FILTER.JBI
	SY_SJ_PICK_CANULA_F_RACK.JBI
	SY_SJ_PICK_FILTER_F_RACK.JBI
	SY_SJ_PICK_SYR_F_RACK.JBI
	SY_SJ_PUT_SYR_T_G1.JBI

The motion element database from the controlling PC is imported into the local memory of the R-Interface and copied to a directory at the beginning of the sample preparation process. The R-Interface 2.0 then receives the first command from the SAMI Ex and enters the relevant task module. The job search algorithm reads the first motion element from the sequence provided by the programmer. Based on the task name, the job search algorithm then taps into the directory list for the relevant task to access the available motion elements. The algorithm then iterates through this list of motion elements and compares each name to the motion element name specified by the programmer. The motion element search is conducted on the basis of the entire motion element name unlike in RS-1 which made the use of node points and key words. A greater advantage of this approach is that motion elements with floating node points can also be detected by the job search algorithm in RS-2. This was not possible in RS-1 as it required matching precise node points to find consecutive motion elements. The directory prevents the program from accidentally calling wrong jobs with similar key words and causing collisions. All motion elements have an inbuilt error message that flashes if the robot arm is not in the correct starting position to ensure operational safety.

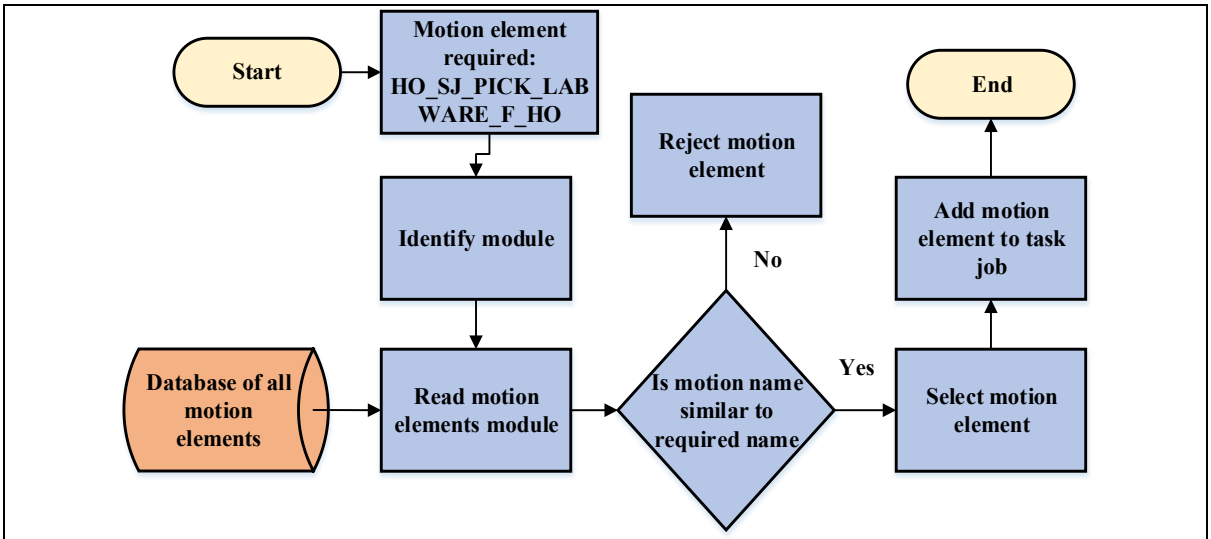


Figure 76: Algorithm for finding motion elements in a task according RS-2

The job search algorithm in RS-2 has an advantage over RS-1 as it will provide a shortcut to find an object whereas in the earlier method the interface program had to look through the entire database serially to find the correct object. All the code names and values are mapped once at the beginning of the process and retained till the end. The motion element database thus need not be copied for each new task reducing by half the latency of a function which is run most frequently in the R-Interface 2.0.

9.1.2 File Transfer Functionality

For a successful sample preparation, it is necessary that all the motion elements are present on the FS100 controller as well as in the motion element database of the controlling PC. The motion elements created on the teach pendant and stored on the FS100 controller are not automatically transferred to the database on the controlling PC. However, that can be done by a single mouse click via an easy graphical interface built in the RS-1. All the motion elements on the FS100 are copied to a new folder in a programmer specified folder on the PC. However, the same is not true for transferring motion elements from the PC database to the controller.

As a large number of motion elements were created for the RS-1, they exceeded the controller capacity and could not all be saved on the FS100 at a time. Hence, all motion elements only for the required tasks had to be manually copied to the FS100. This approach works fine for smaller processes with three or four different tasks. However, for a long process with multiple tasks, if all the tasks have a number of irrelevant files transferred to the FS100, the controller memory filled up fast, and it could not hold all the necessary files.

A number of motion elements were created for specific labware combinations or for specific ALP positions. If these labware, positions, etc. were not being used for the process, the motion elements need not be transferred to the controller. It is however cumbersome to manually sort out the motions and transfer them one by one to the FS100 which sometimes lead to manual errors. It was observed, that wrong motion elements were transferred to the FS100 and some necessary files were missed. This leads to error messages during the process. The process had to be interrupted for loading the files. This was time consuming and also affected the continuity and accuracy of the process.

In order to overcome these limitations observed in RS-1, the automatic file transfer functionality was introduced in RS-2 (Figure 77). This functionality is a part of the job search algorithm. After the job search algorithm selects the necessary motion element from the database, the R-Interface 2.0 checks to see if the motion element is present on the FS100 controller. If it is not present, the file is copied to the FS100, else the algorithm skips this step. The R-Interface however cannot detect unwanted motion elements from the FS100. To overcome this limitation, at the end of each sample preparation process, all the motion elements get automatically deleted from the FS100 controller. Thus for every new process, the FS100 contains only gripper jobs and other system jobs. The motion elements are loaded in run-time as required. The programmer needs to ensure that the motion element database in the controlling PC is updated with the latest version of motion elements before starting the robotic process. The programmer also needs to ensure that the old version of motion elements is deleted from the FS100 controller.

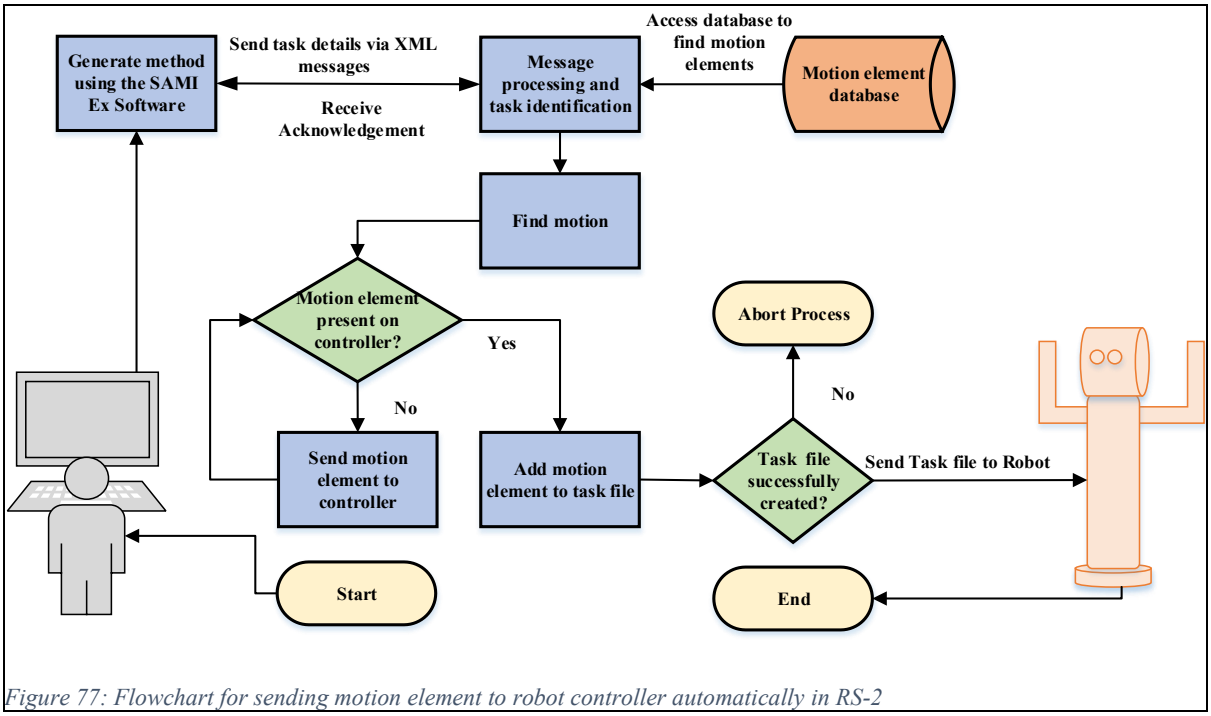


Figure 77: Flowchart for sending motion element to robot controller automatically in RS-2

9.1.3 Error Handling in Remote Mode

The processing of samples in the remote mode is intended to be carried out independently and uninterrupted. However, there may be instances when the robot needs to be stopped mid motion. For example, if the robot task involves putting lids to a 2 mL vial. The lids are small, and if it is not held properly in the fingers, the lid might not fit the vial. This scenario has two outcomes: the lid falls off during vial closing or the vial lid is not securely tight which might lead to vial misalignment or spillage during motion. Both the scenarios can lead to additional errors in the consequent tasks and the sample preparation process may fail. Hence, such errors have to be rectified and the robot process needs to be paused.

The emergency stop button, or opening the robot enclosure door activates a dialogue box on the screen of the controlling PC in RS-1. The dialogue box allows the programmer to pause the process while operating in the remote mode. However, in certain scenarios the robot has a labware in its grip. The robotic process cannot proceed in this case as, the labware might collide with the next motion. Hence it is necessary that the robot is made to release the grip and the arm posture is reset. The R-Interface in RS-1 does not have any solution to this problem.

Hence a manual override button has been introduced in RS-2. In case of an error or impending collision, if the safety door or emergency stop button is pressed, a dialogue box shows on the screen. A 'Manual Move' button has been added to the dialogue box. On clicking the button, the SAMI Runtime software retains its remote connection with the robot. However, the teach pendant can be switched from the remote mode to teach mode and the robot as well as the grippers can be moved using the pendant. The labware can be released and the robot can be put to a convenient position so that the next motion can be started without any errors.

9.2 Motion Planning Algorithm

The RS-2 implements the newly structured and shortened motion elements. They are planned such that multiple tasks can be performed continuously without having to go back to the intermediate or standby position. However, changing the motion elements alone is not enough for a flexible task planning. The sequence of motion elements called by the motion planning algorithm in the robot interface also needs to be optimized to avail the benefits. Thus, the motion planning algorithm needs to be changed in the R–Interface 2.0 to better suit the goals.

Figure 78 depicts the motion planning algorithm in RS-1. The interface program ensures that the robot is in standby position before starting the process else raises an error message. The R-Interface receives commands from the SAMI Ex Runtime software. The commands contain task specific information. The R-Interface access the code section for the particular task to find and combine motion elements into a task job. The task job is created according to the method explained in the previous section. Each task starts and ends at the standby position and the motion elements in RS-1 are programmed accordingly. A new task can be started only after the completion of the previous task. The process is complete after all the tasks in the process schedule have been carried out.

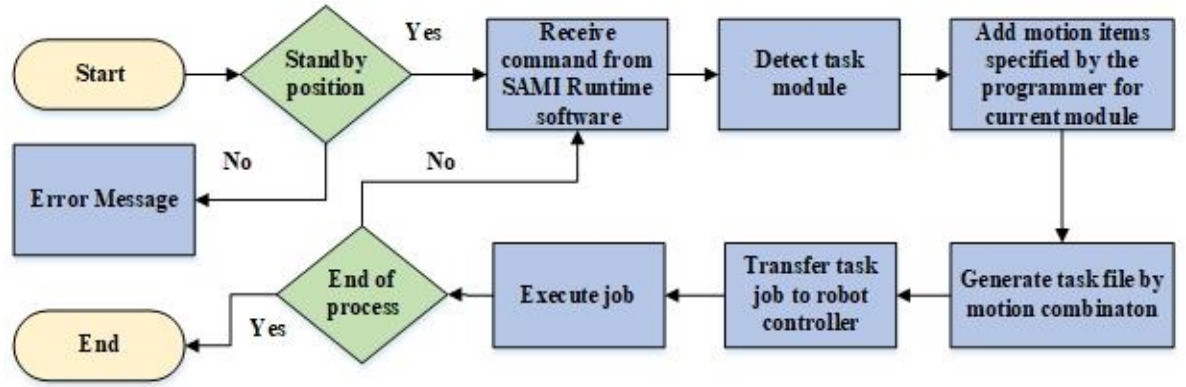


Figure 78: Motion planning algorithm in RS-1

Figure 79 shows the task flow of an example process as done in the RS-1. The process involves five tasks, 1: transportation task, 2: pipetting task, 3: pipetting task, 4: transportation task, 5: transportation task. It can be seen that the robot goes back to standby position after every task. Thus, the arms have to be brought down to the workbench at the beginning of each task. This motion serves as posture adjustment and is time consuming in a lengthy process.

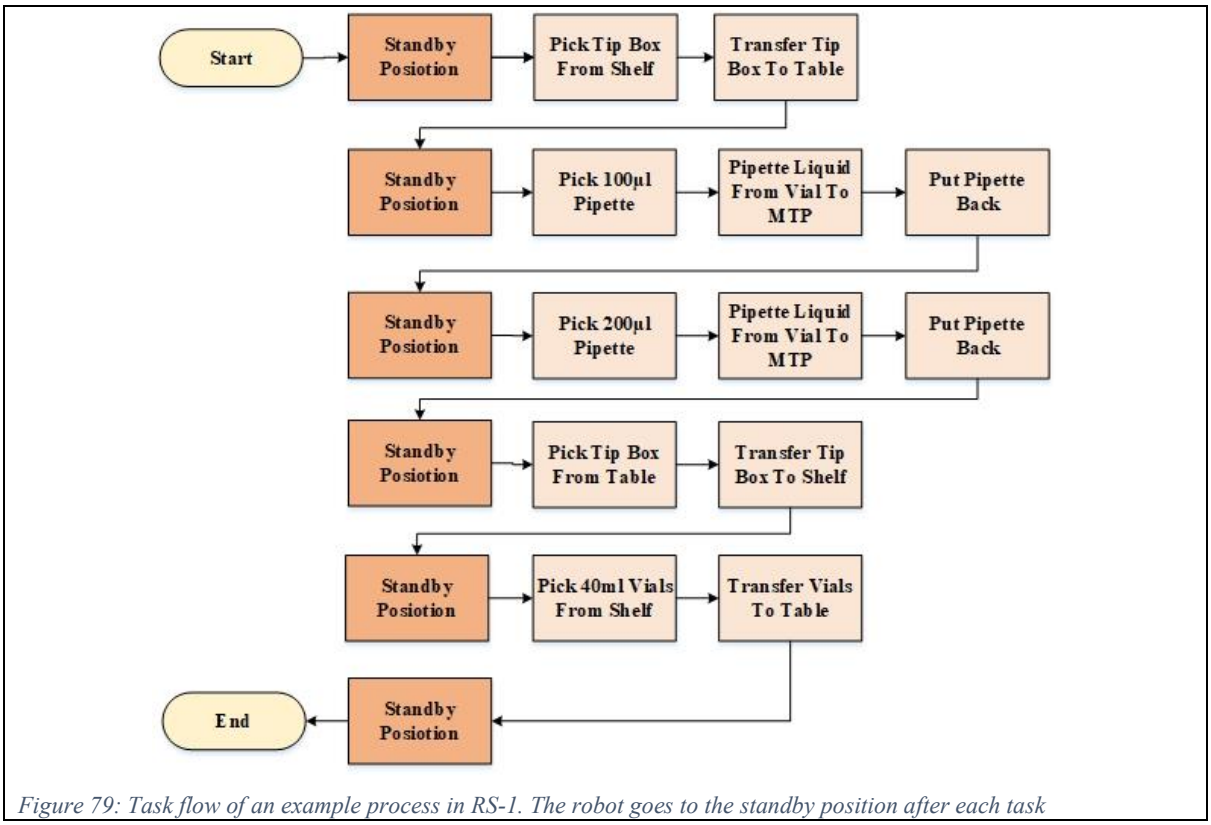


Figure 79: Task flow of an example process in RS-1. The robot goes to the standby position after each task

Thus, the motion planning algorithm was changed in RS-2. The sequence of motion elements for each task is provided by the programmer. However, the algorithm can decide, based on the arm posture at the beginning of the task, whether the arm should move to the intermediate position or not in order to begin the task. If not, the algorithm decides which motion element, out of the sequence of motion elements, the task should start from.

The R-Interface 2.0 commands the robot to move from the standby position to the intermediate position at the beginning of the process. The R-Interface 2.0 then receives the first task information from the SAMI Runtime software. The task command is sent to the message processor by the communication module. The message processor decodes the task message and the particular task module is accessed. At the beginning of the task, the robot checks its current position based on the previous motion element executed. If there has been no task executed yet since the start of the process, the robot detects the position as intermediate position. The task received from SAMI Ex will be considered to be a new task and the motions will start according to the sequence provided by the programmer. The various motion elements will be listed and a task job will be created, the task job will be sent to the FS100 controller through the communication module, via Ethernet in the form of a .JBI file. The task will be executed by the robot. Every task ends either at one of the four node points or the intermediate position. The end position of a task is decided upon which point is closest to the end node of the last motion element of the task. During the robot teaching, direct motion elements are created between the intermediate position and each of the four node points.

The SAMI Ex then sends the command message for the next task to the R-Interface 2.0. The R-Interface 2.0 then refers to the task name of the previous task performed and the key point it ended at. If the end key point of the previous task is different from the starting key point of the new task, the R-Interface 2.0 will call the motion element that will take the robot arm back to the intermediate position. The robot arms will move from the intermediate position to the key point of the new task and the task will be carried out.

In contrast, if the start key point of the new task matches with the end key point of the previous task, the program will directly execute the task jobs. This functionality is of importance when a particular task is repeated or the places of two labware are exchanged. The robot need not move to the intermediate position in between two identical tasks. The common key points are of importance in this scenario. The start and end points of task are the same; hence, the robot can repeat the motions without going back to the standby / intermediate positions or changing its posture. Changing position from one posture to another is tedious and it is faster to go via the intermediate position. The process is finished when all the tasks from the process schedule have been performed. After the last task, the robot arms move to the standby position indicating the end of process. Figure 80 graphically depicts the motion planning algorithm in RS-2.

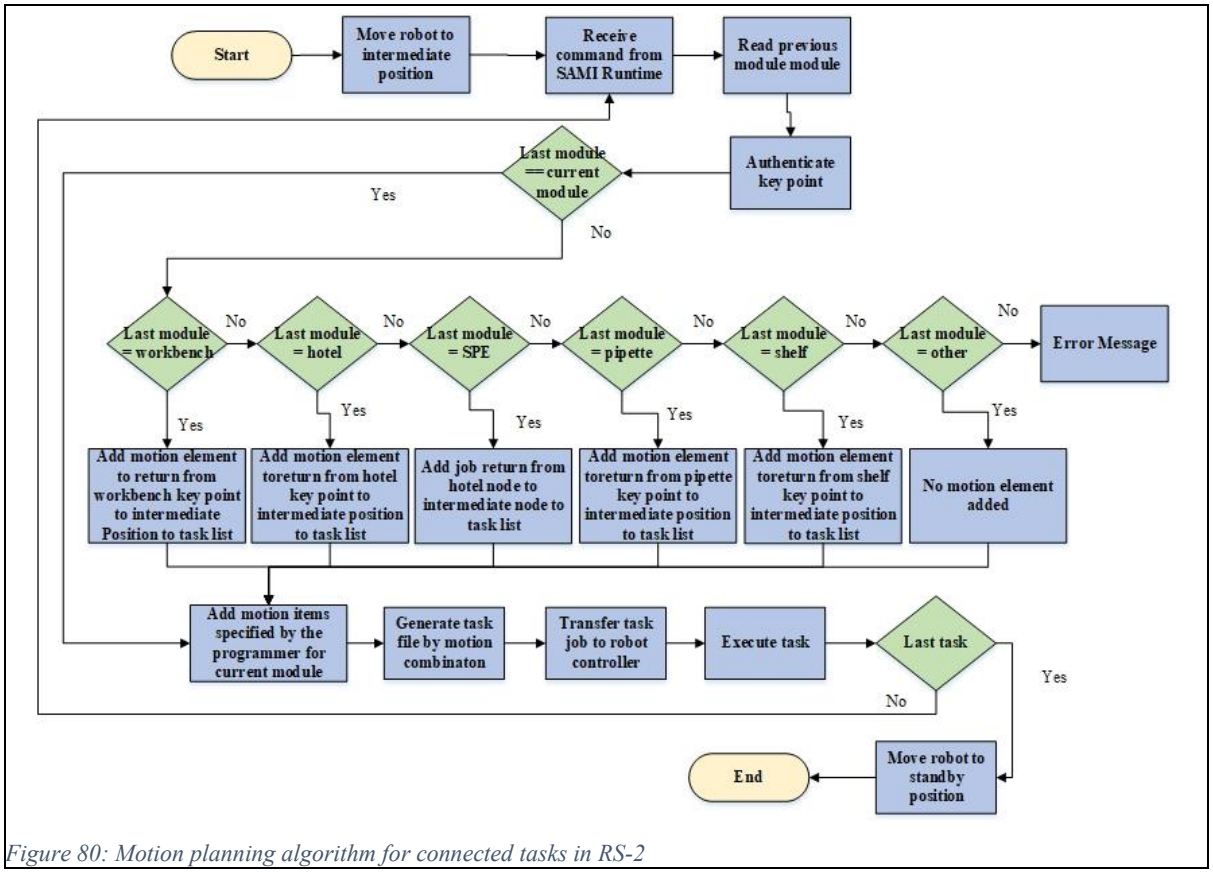


Figure 80: Motion planning algorithm for connected tasks in RS-2

In order to observe the effect of the motion planning algorithm, task flow of the example process (described in Figure 79) as performed in RS-2 can be seen in Figure 81.

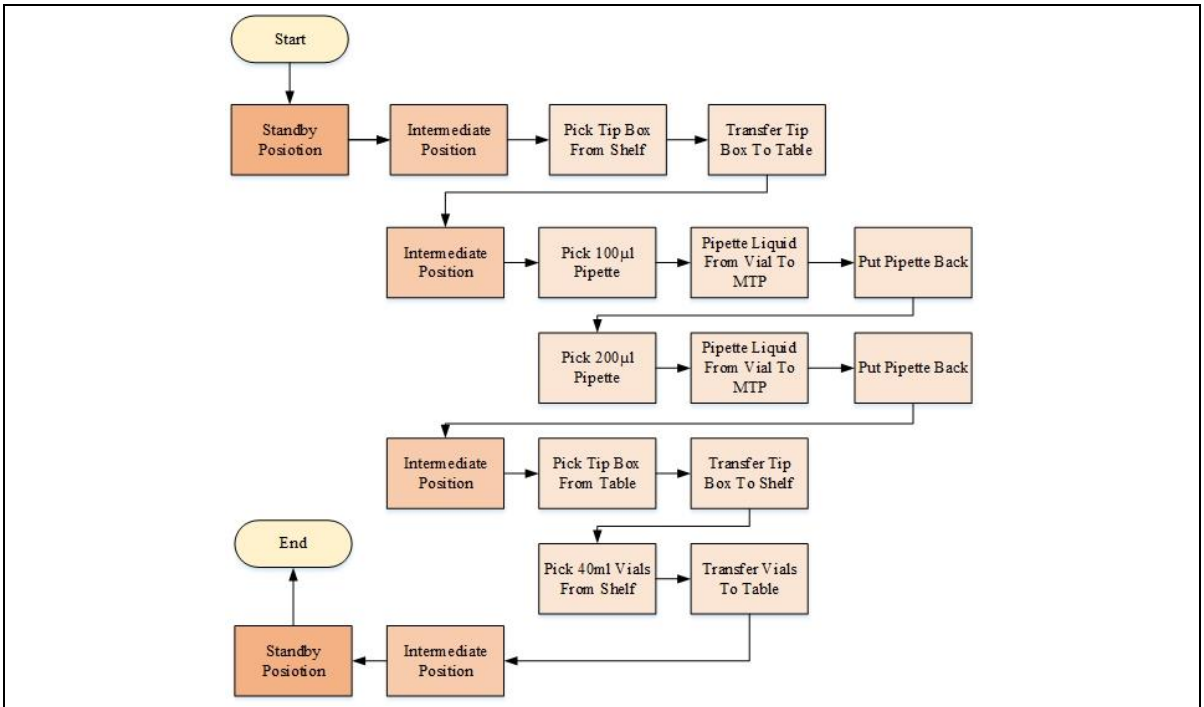


Figure 81: Task flow of an example process following the path planning approach in RS-2. The robot goes to the intermediate node only when necessary.

It can be seen, that the robot moves from standby position to the intermediate position at the beginning of the process. The first transportation motion is carried out which ends at the workbench key point (Figure 82a). The robot arm cannot move directly to the pipette key point to pick up the pipette due to limitation of joint angle rotations. Hence, the arms are moved to the intermediate position. At the end of the first pipette task, the robot arm is at the pipette key point (Figure 82b). The next task is also a pipette task, which can be started at the pipette key point and hence the task is continued without the robot moving to the standby/intermediate positions. The fourth task which is a transportation task again needs the robot arms to be brought to the intermediate position. The fifth task, which is transportation again is carried out continuously. At the end of the process, the robot arms are brought back to the intermediate position form the workbench key point and finally moved to the standby position. The process is finished and the robot can be stowed in this position until the start of the next process.

Figure 82: Robot arm position in (a) workbench key point (b) pipette key point

9.3 SAMI Method Creation

As previously seen in Chapter 3, individual SAMI Ex methods were created for each task in RS-1. Consider an example process containing six tasks. An adaptor with 12 vials containing raw samples is placed on an ALP and is pipetted with a reagent. The sample reagent mixture is then filtered using a standard PTFE filter attached to a syringe. The filtered liquid is diluted by pipetting a diluting agent. The 12 vials are then lidded and transferred to the thermoshaker for incubation. Finally, the vials are transferred to the LC for analysis. Thus, 6 SAMI Ex methods need to be created for one sample preparation process. The six method files are queued in the right sequence in the SAMI Runtime software during program execution to form the entire process. However, this kind of process design is inconvenient for preparing multiple batches of samples continuously. A 'batch' is the number of times the sample preparation process is repeated uninterrupted. In case of multiple batch processing, the queue of the six method files will have to be repeated for each batch. The source labware positioning for each batch is different. Hence, new method files will have to be created for each task of every batch giving rise to a large number of files to manage. This type of SAMI Ex method creation is time-consuming and may introduce positional errors or discontinuities in the process flow possibly leading to damage to labware and error messages.

Prior to processing, the SAMI Runtime software validates the positioning of labware on the workbench. Labware in all of the methods has to be placed on the workbench, thus the SAMI Runtime detects it as ALP sharing by multiple labware which is not permitted. This results in an error message (Figure 83) and the programmer cannot start the process. Thus, it is not possible to queue the sixty task files for continuous sample preparation and batch processing cannot be done.

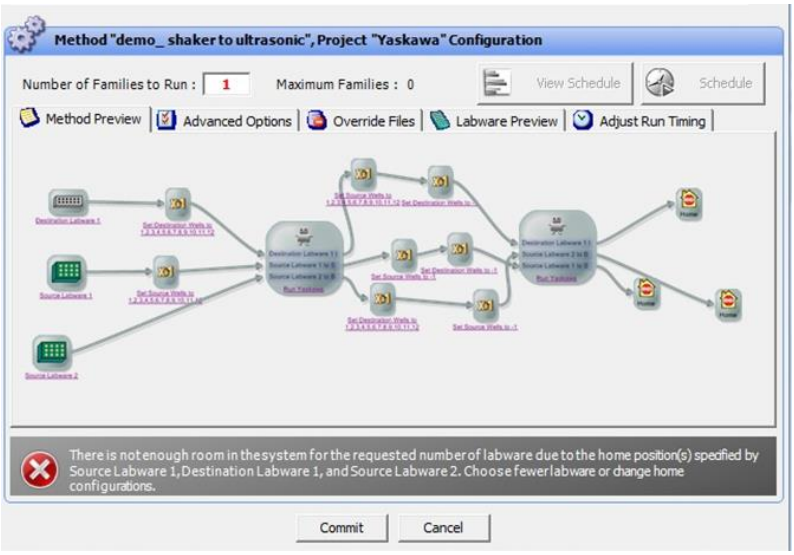


Figure 83: Error message by the SAMI Runtime software for duplicated labware positions in a multi-method queue

As RS-1 was equipped with creating only one batch of samples due to the way the SAMI Ex methods had been created, the processes started with most of the labware being manually

placed on the workbench at the beginning on the process. Thus, certain motion elements for transferring labware such as syringe filters, cannulas, vial lid racks, MTP lids between the storage racks and the workbench were not programmed. These motion elements are however necessary for creating multiple sample batches, as the labware needs to be replaced and replenished as required.

In order to reduce the effort of creating and troubleshooting individual SAMI Ex methods, the process flow is created as a single method file in RS-2. Figure 84 shows the consolidated method with the entire process [118]. The consolidated method lets the programmer view the complete process at a glance. Any change in one of the tasks is reflected in the consecutive steps. Hence, process planning errors and discontinuities can be traced easily without having to access multiple files. Tasks can be added to edit in case the measurement procedure changes.

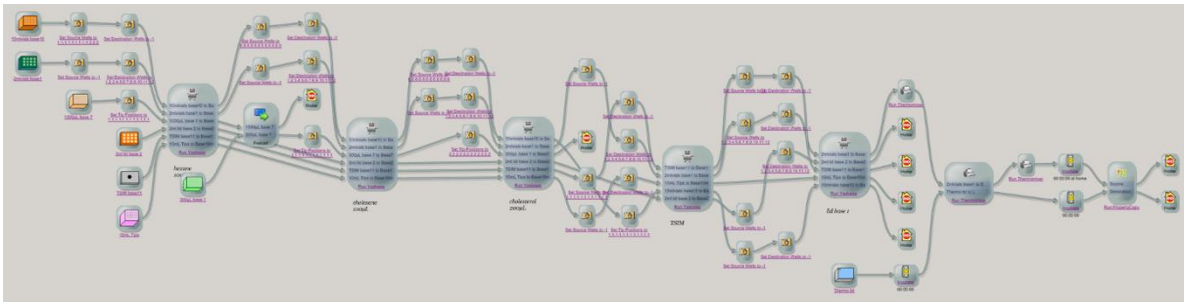


Figure 84: Method file in SAMI Ex software for sample preparation process for determination of cholesterol in biliary stent incrustations.

The consolidated method also allows the system to create multiple batches of samples using only one SAMI method file with the help of 'Families'. A family is a group of all labware required for one batch. To understand families better, it should be assumed that the robot prepares four sample vials in a process. An adaptor consisting of four vials is placed on an ALP, all four vials together are considered as one labware and hence one family. For the robot to prepare three batches, three such families (adaptor with 4 vials each) are required to be provided to the robot (Figure 85). An adaptor has the capacity to hold 12 vials. Hence even if an adaptor has the capacity to hold 3 families, the SAMI Ex method cannot identify them.

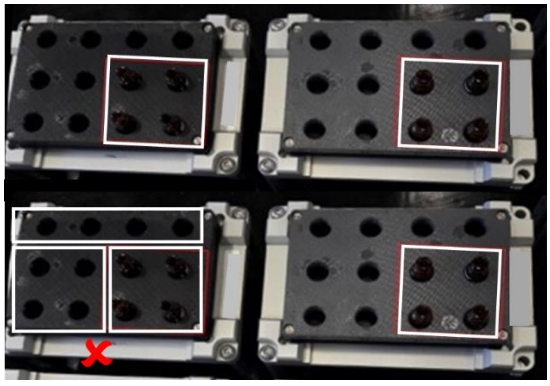


Figure 85: Three families of vials placed on the ALPs. The three families cannot be placed on one ALP.

The number of families defines the number of batches the robot is capable of preparing as unique initial positions of all labware required for all batches has to be specified while creating the SAMI Ex method. The batch capacity of the robotic system is unique for each sample

preparation process and depends upon how long or complicated the process is. The approach of a consolidated SAMI Ex method using families makes batch processing possible as only one method is created and made to run on a loop. For each new iteration, the robot uses labware from a new family. The labware processing is common for all the batches, only the initial positions of the labware are variable.

Labware in a SAMI Ex method can be defined in two ways: as source or a resource pool. Depending upon the labware type, labware can be reused for multiple batches (saving space and labware costs), or new labware might be required for each batch. For example: if one batch requires 10 pipette tips of 100 µL volume, one tip box can be reused for at least 9 batches, as the tip box can hold 98 tips. Such labware is defined as a 'source' in the SAMI Ex method. A reservoir containing reagents in large quantities can also be defines as a 'source'. The source may also contain multiple labware. The SAMI Ex method can be so designed to reuse a labware until it is empty and then exchange it for a full labware from the source. However, plastic syringes, filters, cannulas, vials, etc. cannot be reused. Hence, 9 different syringe, filter and cannula adaptors will be required for 9 batches. Such labware is defined as a 'resource pool'. Typically, stock solutions to be analyzed, labware holding reagents during the process are defined as resource pools. A new labware from the resource pool is used for each family. Thus, a single SAMI Ex method contains a mixture of labware define as sources or resource pools.

A SAMI Ex method with families is run via the SAMI Runtime Software. As the families are already integrated in the method, only one method is queued in the SAMI Runtime Software. The progress of the automated process can be seen on the screen of the controlling PC. The positions of all the labware sources and resource nodes are depicted as color coded icons. Each icon is designated a number which signifies the family it belongs to. Thus, if the second family is being processed, all labware marked with the number 2 will be transported to the workbench during processing. The labware can be seen flowing between the various position as the process advances. Figure 86 depicts the labware storage in the robot environment as seen in the SAMI Runtime software.

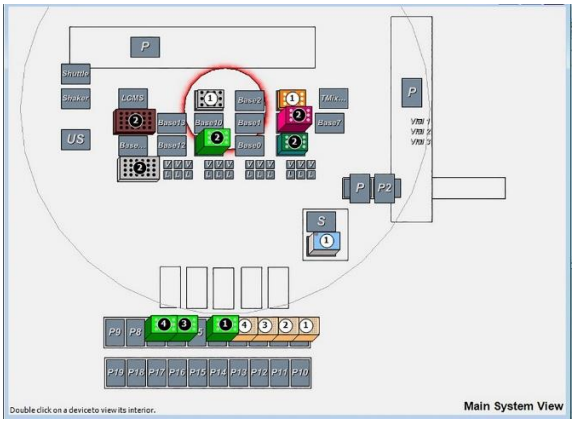


Figure 86: Runtime system view of a running process depicting the various labware and their positions, the family number being run and the ongoing task in real time.

10 Motion Elements in Applications

The motion elements were changed in the new system to suit the new node and key point system. The five node points were used to connect the various tasks with each other. These node points help the robot in smoothly transitioning from one task to the other without going to the standby position. The key points connect the various motion elements required to create one task. In the earlier system, each motion element had a different start and end point. This method was created from the safety point of view so that the robot could not start wrong motions causing collisions. This method however also made it difficult to connect various motion elements together as the start and end points did not match, the interface program could not find connected jobs to create a task sequence. The motion elements could only be carried out in a predefined sequence. Adding new motion elements or connecting existing motion elements to now ones was difficult as a result. The new system overcomes this issue. Table 12 below lists the various tasks and the motion elements required for the tasks.

Table 12: List of all the tasks performed by the robot

Task	Motion Elements
Pipette task	Pick pipette Put back pipette Load pipette tips Get Liquid Out Liquid Release tips
Electronic pipette task	Pick pipette Put back pipette Get liquid Out liquid Load pipette tips Release tips
Vial task	Pick vials Put vials Transfer vials from R1 to R2 Transfer vials from R2 ro R1 Open vial lid Close vial lid
Hotel task	Transfer labware between hotels and workbench Transfer labware between hotels and SPE
Shelf task	Transfer labware between shelf and workbench Transfer labware between shelf and SPE
Syringe task	Get syringe Transfer syringe to R1 Load cnnula Release cannula Load filter Ger liquid Out liquid Release syringe and filter
Glass syringe task	Pick glass syringe Get liquid Out liquid Put back glass pipette
Glass pipette task	Pick glass syringe Get liquid Out liquid Put back glass pipette

SPE task	Transfer labware between SPE and workbench Pipette liquid into labware
GC task	Pick up vials from labware Transfer vial to R1 Transfer vial to R2 Transfer vial to GC
LC task	Pick labware from workbench Transfer labware to LC tray Open LC Transfer tray to LC Close LC Transfer labware from LC tray to workbench
TM task	Open thermoshaker lid Transfer labware from workbench to thermoshaker Transfer labware from thermoshaker to workbench Close thermoshaker
Shaker task	Transfer vials between workbench and shaker Transfer vials between shaker and ultrasonic bath Turn on/off shaker
Ultrasonic bath task	Transfer vials between workbench and ultrasonic bath Transfer vials between shaker and ultrasonic bath Turn on ultrasonic bath

As previously discussed, the motion elements have been optimized. The changes in motion elements and a comparison between the RS-1 and RS-2 for all the tasks is described in this section. The motions are explained graphically as shown in Figure 87. The coloured arrows mark the direction of flow of labware. Figure 87 does not contain a standby position as this posture is not required for the motions.

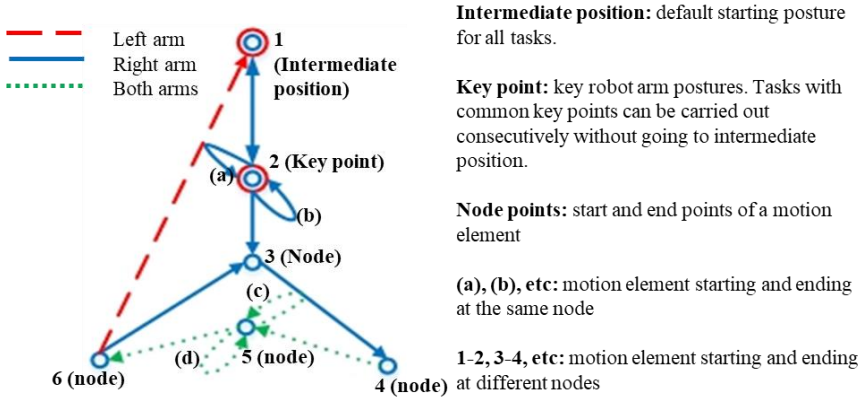


Figure 87: Template for understanding the motion element diagrams detailing the flow of labware, arms used, modes reached and the motions carried out.

10.1 Pipette

Figure 88 is a graphical representation of the pipetting task. All the manual pipettes, single channel as well as multi-channel are hung on a rack with the help of adapters. The right arm picks up the pipettes and loads the tips from a tip box, the left arm actuates the piston for aspirating and dispensing liquid as well as to reject the used tip in a waste bin. The right arm then places the pipette back on the rack.

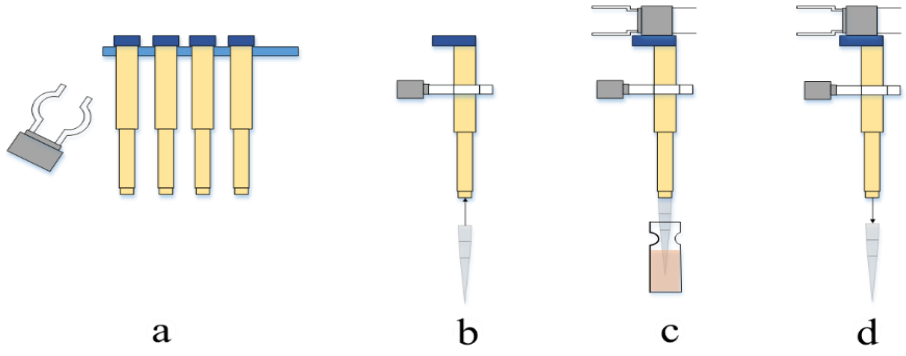


Figure 88: Pipetting task (a) pick/ put back pipette, (b) load tips, (c) aspirate/dispense liquid, (d) reject tip

The motion element diagram (Figure 89) depicts the motions according to the RS-1 and RS-2. The motions in RS-1 start at the standby position. The right arm moves to a prepare position (1-2). The pipette is picked and placed (2-3). The arm then moves to a prepare to load position (3-4). This motion changes the posture of the robot arm to make it suitable for tip loading. RS-1 contains eight motion elements for the eight different pipettes. After loading the tips (4-5) to the pipette, a new motion element brings the left arm from standby into prepare position for pipetting action (1-5). This motion is long and time consuming for processes with multiple pipetting tasks. The motions for aspirating liquid and dispensing liquid are made so that the arm posture for all the ALPs is different. The robot arms have a defined start and end positions for these motions (motions 6-7, 7-6). The arms move to these positions between consecutive pipetting motions making them look repetitive. All the motion elements have a different end point, which was designed for safety. However, change in the terminal position of one motion element introduces discontinuity, which is difficult to detect.

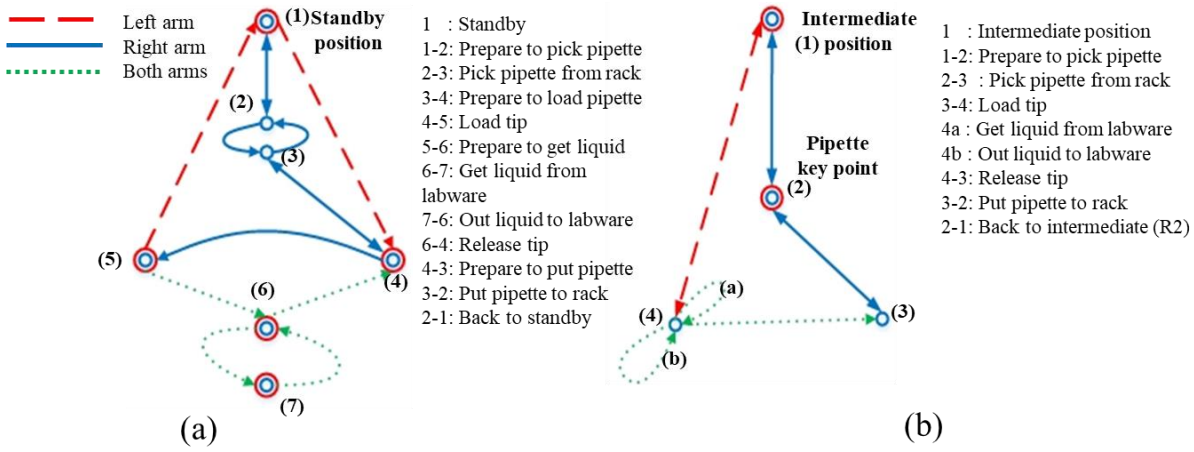


Figure 89: Motion elements for manual pipetting tasks (a): according to RS-1 [111]*, (b):according to RS-2*

The motions in RS-1 were edited to change the starting position of the task and re-fragment the task. A 10 µL single channel pipette was added and suitable motions were programmed. Motions for the first (unprecedented by a similar task) pipette task in RS-2 begin at the intermediate position (1). The right arm moves to the pipette key point (2). After picking up the pipette, the robot comes back to node 2 unlike in RS-1 where the motion ends at a different node. Thus, from node 2 in RS-2, the robot can also put the pipette back if required. This way a pipette pick and put can be performed consecutively making the motion retraceable. This feature in RS-2 is important as it allows two pipetting tasks to be carried out consecutively

without having to go back to the standby position. Motions (2-3) and (3-2) in the figure depict the motions of picking and placing back the pipette. The robot arm moves to node 3 (equivalent to node 4 in RS-1) to reach a posture conducive for loading tips. The robot then picks up the tips during motion (3-4) and comes back to the node point 3, unlike in RS-1, where the motion ends at a new node. During the same motion, the left arm also moves from intermediate position to node 4 (RS-1 contains an independent motion to change the posture of left arm). Thus, two motions from RS-1 have been shortened into one synchronous motion in RS-2. RS-2 has only one motion element for all tips from 1 mL-10 µL volume. 10 mL and 5 mL pipettes have individual load tip motion elements. The posture in node 3 is adjusted to be compatible for all ALP's so that individual programming is not required. Both the arms move synchronously to reach each of the positions. The node point (4) depicts a floating node. The arms do not come back to a defined position, instead the motion ends after the pipette exits a labware. The motion elements (4a) and (4b) depict the aspiration and dispensing liquid and do not have a fixed start and end point. After releasing the tips, the right arm comes back to node 3 and left arm goes to intermediate position in a single motion. A new motion element is not required for left arm as in RS-1. The robot can now either put the pipette back to position or load a new tip. As can be seen from Figure 89, RS-2 contains a smaller number of nodes than in RS-1. Multiple motion elements share nodes, which helps in maintaining continuity and simplifies motion editing. The main task motions all merge at the node point 3. In the eventuality of a more intelligent system with feedback, the robot can discard damaged tips immediately after loading. In case the grippers are changed to multi finger (human like finger-thumb) design, the existing motions can be used as is for a single-arm manipulation task. The pipetting task ends at the pipette position after the used pipette is put back in place.

10.2 Electronic Pipette

The task of electronic pipettes was newly added, the graphical representation of which can be seen in Figure 90. The two electronic pipettes (1,000 µL and 200 µL volume) are attached to the communicating modem with a connecting wire and are hung on a rack above the robot. The motions of the task are kept similar to a manual pipetting motion, except only the right robot arm is utilized as can be seen in the Figure 91.

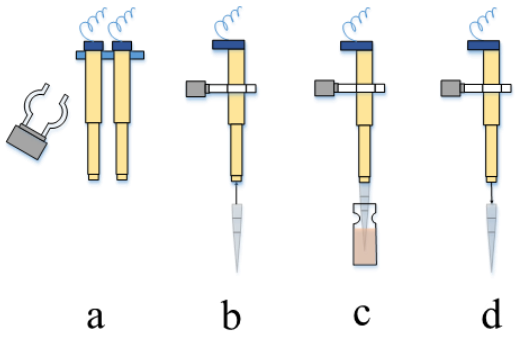


Figure 90: Electronic pipettes. (a): pick / place pipette, (b): load tip, (c): aspirate/dispense liquid, (d): reject tip

The node points used for the motion elements are same as those used in manual pipetting task. Thus, the manual pipette motion elements can be easily adapted for electronic pipettes as the position of tip boxes and vials on the workbench is similar.

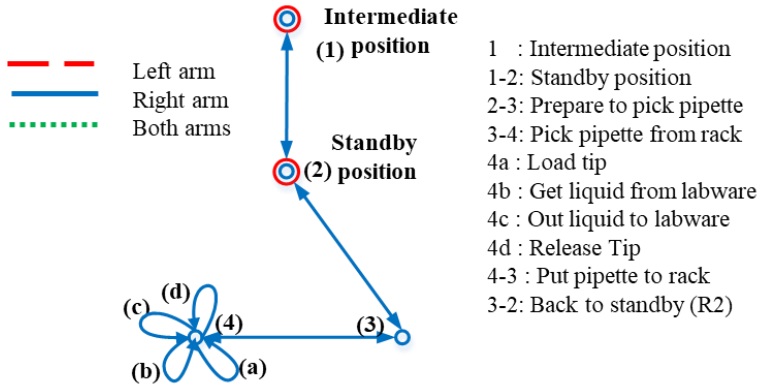


Figure 91: Figure depicting the motion elements for the electronic pipette task in RS-2

The robot moves from the intermediate position to the standby position (1-2) at the beginning of this task and task initiates with the robot in the standby position so that the left arm does not obstruct the motion of the right arm. The left arm is not used throughout the task. Motion (2-3) depicts the arm travel from standby position to the pipette rack. The arm is fully extended vertically in this position. The robot can access both the pipettes from this node. The robot picks up the pipette and moves to node 4 to repositions itself above the workbench in order to load the tips. Node point 4 in the electronic pipette task coincides with node point (4) in manual pipette task. The motions to load tips, aspirate liquid, dispense liquid and release tip, as depicted by motion (4a), (4b), (4c), (4d), start and end at the node point 4. At the end of the liquid transfer, the pipette is placed back on the rack as depicted by motion (4-3). From here, the pipette may be picked up for a consecutive pipetting task or the arm may go back to the standby position (3-2).

10.3 Glass Pipette

The glass pipette is placed in a specially designed rack on the workbench (Figure 92). The rack can hold up to 12 vials in a 4x3 configuration. The right arm picks up the glass pipette while holding to the glass body. The left arm presses and releases the rubber bulb to aspirate and dispense liquid. After use, the glass pipette can be placed in the rack for reuse or placed in a new waste rack. The waste rack is similar in design to the pipette rack.

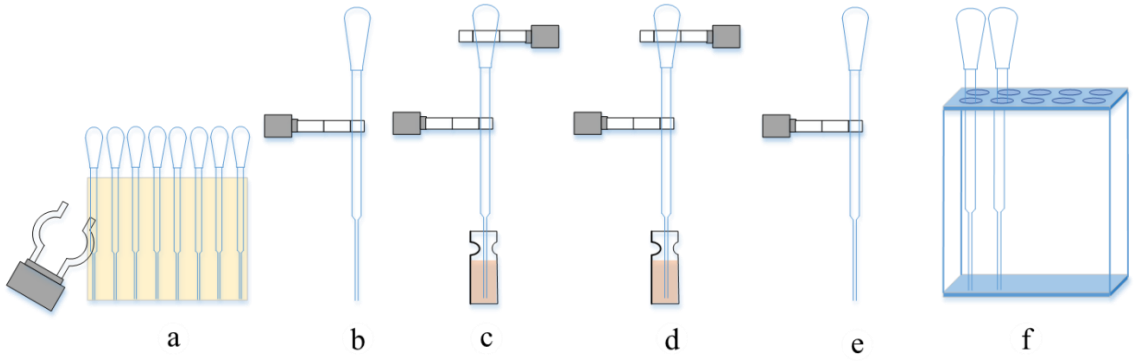


Figure 92: Glass pipette: (a) prepare to pick pipette (b) pick pipette, (c) aspirate liquid, (d) dispense liquid, (e) put pipette, (f) reject pipette

The motions in RS-1 started at the standby position (Figure 93a). The right arm is brought to a prepare position (motion 1-2). The glass pipettes are picked up by the right arm (2-3) and taken to a prepare position (3-4). The left arm also moves from the standby position to the prepare position. The get liquid and out liquid motions have a fixed start and end points (4-5) which are non-identical making the motions non-retraceable. After the pipetting action the arm moves back to the prepare position (4-6). It should be noted that node 2 and node 6 are non-identical even though the arm is in prepare posture. The pipette is then put in the appropriate position (6-2). Both the arms return to the standby position (2-1).

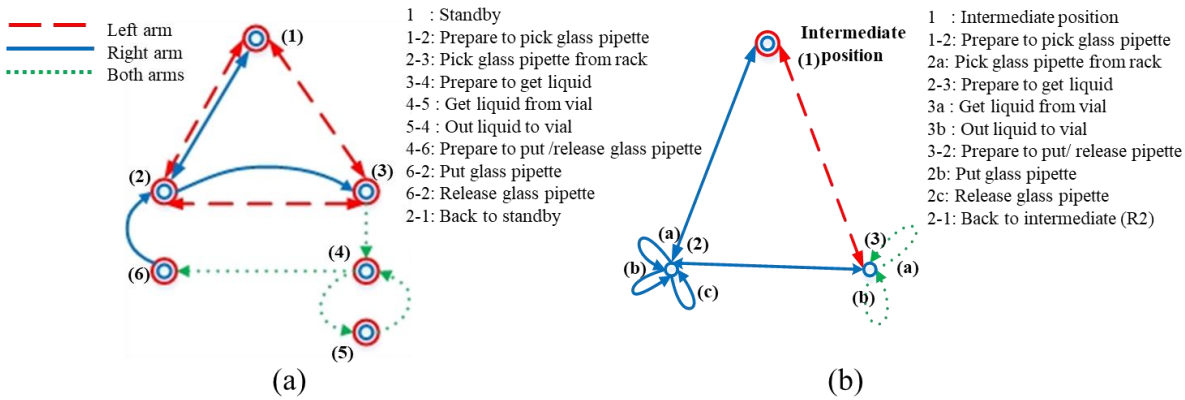


Figure 93: Figure depicting the motion elements for glass pipette task (a):according to RS.1 [111] *(b): according to RS-2*

The motions of the task are largely unchanged as compared to RS-1 except for the end positions of the motion elements. The glass pipette task in RS-2 starts from the intermediate node as seen in Figure 93b. Motion (1-2) positions the arm to pick up the pipette. This is a posture adjustment motion. However, this motion has not been eliminated in RS-2 as it can be

combined with other motion elements to access manual pipettes hanging on the temporary stand. At node point (2), the robot arm is in a position to get a new pipette, place a used pipette back in the rack for later use, or place the used pipette in a specially designed waste rack. The motions are depicted by (2a), (2b) and (2c). The dual-arm robot picks up the glass pipette with its right arm. The left arm in intermediate position is already in a prepared state. Both the arms then move to node point (3) as shown by arrows (2-3) and (1-3). The dual arms work in coordination to aspirate and dispense liquid depicted by motions (3a) and (3b). The motions start and end at the common point allowing continuous motion. The glass pipette is held in the right arm, while the left arm pinches the rubber bulb to aspirate and release liquid. The glass pipettes can be used on any of the ALP positions and with any vials like a normal manual pipette. The motions (2-3) and (1-3) are retraceable. That means in case of errors, the robot arm can be jogged backwards to a previous point to rectify the error. At the end of the task, the left arm goes back to the intermediate position following paths (3-1). The right arm moves to node 2 to reject the pipette and travels back to the intermediate position at the end of the task. The motion element in RS-1 can be combined in a unique way to create a task as each motion element has a unique start and end node. Unlike RS-1, the glass pipette task in RS-2 has only three node points. The common end points allow the motion elements to be performed in varying combinations if required.

10.4 Shelf transfer

Figure 94 graphically depicts the transfer motion of labware from the horizontal shelves to the table. The left arm grips the labware such that robot fingers are parallel to ground. The robot however cannot place the labware on the table while holding the labware this way. Hence, the labware is placed on a regrip station (a raised ALP specially designed for this purpose) and the grip angle is changed as can be seen in Figure 94c.

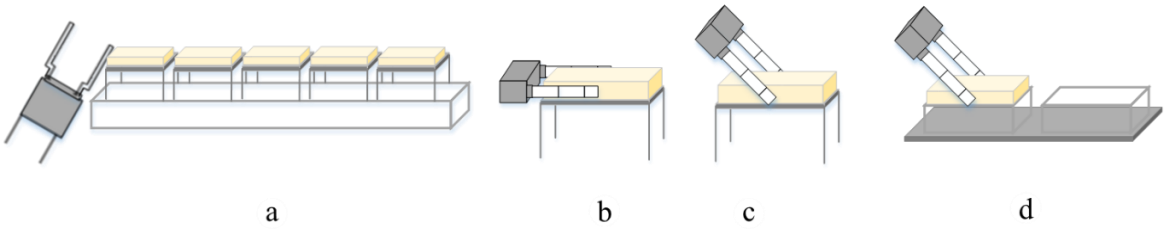


Figure 94: Transfer from shelf: (a) pick labware, (b) place on elevated ALP, (c) change grip position, (d) place labware on workbench.

The right arm in RS-1 started at the standby position and moved to the prepare position (motion 1-2 in Figure 95 a). The motions for get labware (2-3) and put labware (3-2) start and end at varying points. The labware is placed on the regrip station (3-4). The arm with the labware travels to the workbench where the labware is set down (4-5). The arm travels back to the standby position at the end of the task (5-1). The task was programmed only for transferring tip boxes from the shelf. For multiple transfer motions, the arm had to start the motion from the standby position.

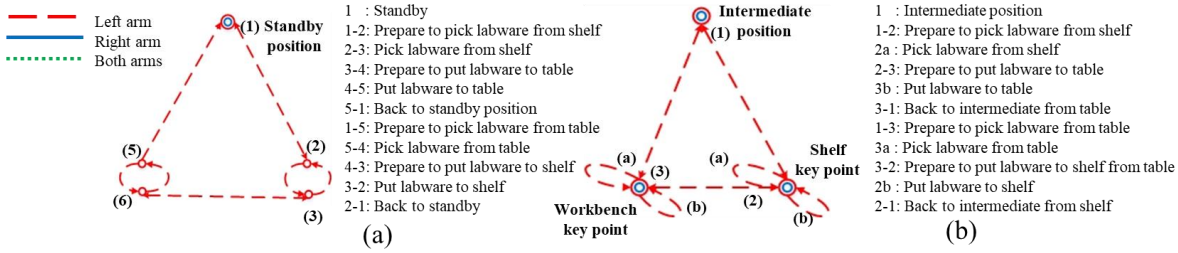


Figure 95: Motion elements for labware transfer between shelf and workbench (a according to RS.1 [111] *(b): according to RS-2*

The shelf transfer motions have changed as compared to RS-1 (Figure 95b). The motion elements of the first shelf task (task unprecedented by a shelf task) start at the intermediate position instead of standby position as in RS-1. An entirely new motion was designed to achieve this. The arm motion 1-2 in RS-2 is 1.5 times faster than arm motion 1-2 in RS-1. The left arm changes posture to be suitable for picking labware from the shelf (1-2), thus reaching the 'shelf key point'. Consecutive shelf tasks start and end at this point. The motion elements (2a) and (2b) depict the put labware and pick labware motions and the arm returns to the key point. Both the tasks can be performed consecutively in a loop. Consecutive pick and place tasks are made faster this way as both motions start and end at the same position and the robot does not need to change postures. A new shelf for storing labware was added on top of the existing shelf with a view to optimizing vertical space in RS-2. New motion elements for picking and placing labware were programmed for this shelf based on the same principles extending the functionality and adding new storage spaces. This added space is useful while performing batch sample production. A special variable is assigned to each shelf for identification such that the robot can differentiate between 'position 0 on shelf 0' or 'position 0 on shelf 1'. The left arm picks the labware and travels to the 'workbench key point' in motion (2-3). The grip change is done during motion 2-3. The pick and place labware to workbench motion elements start and end at this position. The motion elements (3a) and (3b) depict the picking and placing labware at the workbench. The positions of labware can thus, be changed, if necessary, without the robot arm moving back to the intermediate position. After placing a labware on the shelf, a new labware can be picked up from the same position, without the arm moving to the intermediate or standby positions. The motions (3-2) depict the motion of the arm from the workbench to the shelf whereas (2-1) depicts the movement of arm from the shelf to the intermediate position. The motion (3-1) depicts the movement of the arm from workbench to intermediate node. By default, the task ends at the workbench key point or the shelf key point depending upon the direction of labware transfer. All the motions are retraceable so that the arm can be taken back to the previous position if necessary, during the motion. The task was programmed in RS-2 for a variety of labware, like transporting SPE cartridges, syringe racks, 22 mL, 10 mL and 40 mL vials. 2 mL and 4 mL vials are typically stored in hotel shelves although they can also be stored and transported from the shelf if required and are compatible to the motion elements in the shelf task module. These labware are heavier than the tip boxes and require a different gripping position for optimum stability during motion. The node points are reduced in comparison to RS-1.

10.5 Hotel transfer

A graphical depiction of the hotel task is shown in Figure 96. The left arm is used for this task. The robot can transfer labware from the hotels to multiple destinations with the hotel task. The left arm picks up the labware from one of the hotels. Depending upon the destination, the grip angle is changed. The labware is placed on the regrip station for this purpose.

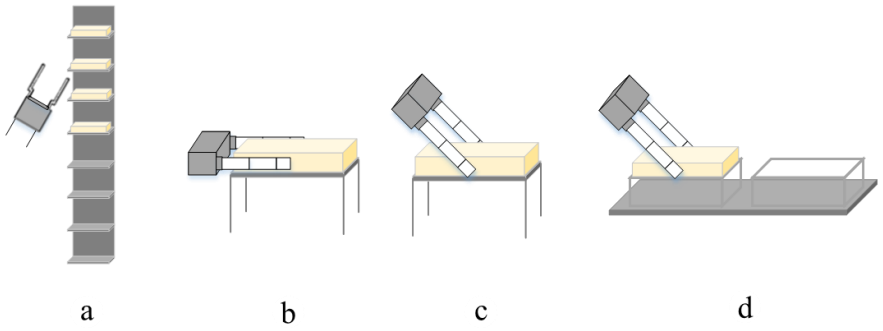


Figure 96: Hotel transfer (a): Pick labware, (b): place on grip changing station, (c): change grip, (d): place on workbench

Figure 97 depicts the motion element for the task of labware transfer from the hotel. The motion sequence in RS-1 started from the standby position. Only the left arm (R1) is used for the task. The arm moved to the prepare position in motion (1-2). The labware was then picked from the hotels and came back to node 3. The start and end positions of the labware pick and place motions was non-identical. The arm then changed the posture to a prepare position at node 4 and transferred the labware to the workbench (4-5). The arm returned to standby position at the end of the motion. The next hotel task again started at the standby position. The RS-1 consisted of three hotel structures. One hotel structure can hold 8 labware simultaneously. However only the top four shelves in each hotel had been programmed in RS-1. Separate motion elements following the same path were created to transfer MTPs and reservoirs.

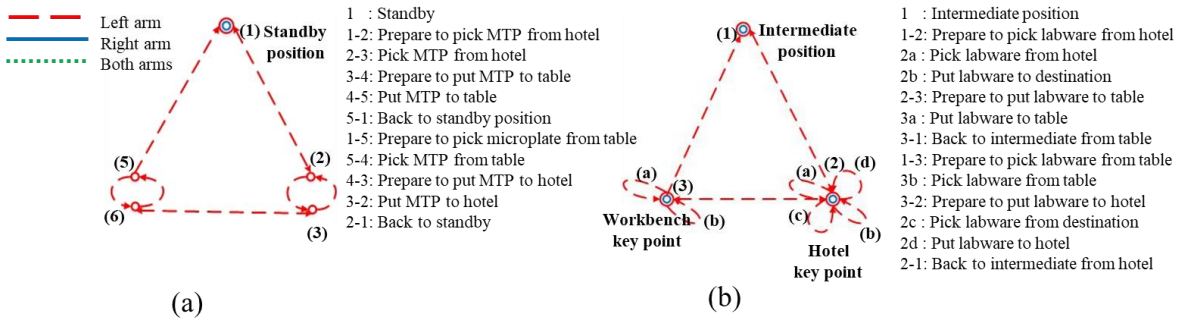


Figure 97: Motion elements for transferring labware from vertical storage to workbench; (a): according to RS.1 [111] *(b): according to RS-2*

The hotel motions were expanded to include more capabilities (Figure 97 b). Two new hotel structures were introduced and motion elements were programmed to service them. The existing motion elements in RS-1 were expanded so that the lower positions (position 6 – 8) in hotels could also be accessed by the robot. The new additions expanded the storage space and utilized workspace more efficiently. Due to the design of the hotel structure, the first five labware can be accessed with the robot arm parallel to the workbench. The later positions have to be accessed at an incline due to space limitations. The robot motion in RS-2 starts from the

intermediate position (1). The left arm (R1) moves to the hotel key point (2). The SPE task can be connected to a hotel task through this key point. The motion 1-2 is retraceable and the arm can travel back to the intermediate position following motion (2-1) if required. The motion 1-2 in RS-2 is twice as fast as motion 1-2 in RS-1. At node 2, the robot can be commanded to pick a labware (2a). New functionalities are added in RS-2 such that the robot can transfer filter and cannula racks directly to designated ALPs on the workbench as well as labware to SPE carriage (2b) without having to change the grip angle. The robot arm returns to the hotel key point after this motion. In order to transport labware to the workbench (2a), the grip angle is changed by placing the labware on the regrip station (2-3). The motions start and end at the same point hence can be performed consecutively. This feature can be utilized if a wrong labware is picked up in the hotel. The arm then moves to the workbench key point (3) where the labware is placed (3a). Motion (2-3) and (3-2) allow the arm to move bi-directionally between the hotels and the workbench. The task ends at the workbench key point. A motion element to return the arm to intermediate position is created (3-1) which is used optionally depending upon the next task. At the workbench key point the robot can pick up a labware from the table (3b) and move to the shelf key point (2). The labware can be transferred to the hotel (2d) or to the SPE. The arm returns to the hotel key point after placing the labware where the task ends. RS-2 allows the robot to transfer vial lid racks, reservoirs and MTPs with lids from and to the hotels which could not be done in RS-1. No special motion element is required for this task. The robot can transfer 2 mL vials, 4 mL vials and 10 mL vials as well. All the labware vary slightly in terms of the gripping force required or the position of grip on the labware. This decision is programmed in the motion elements with the help of variable indices and depends upon the type of labware being transported.

10.6 Syringe

The dual-arm robot is capable of using a general-purpose syringe required for important tasks of filtering the sample solutions. Both the arms are required for this task as seen in Figure 98. The syringe is held by the left arm while plunger actuation is done by the right arm.

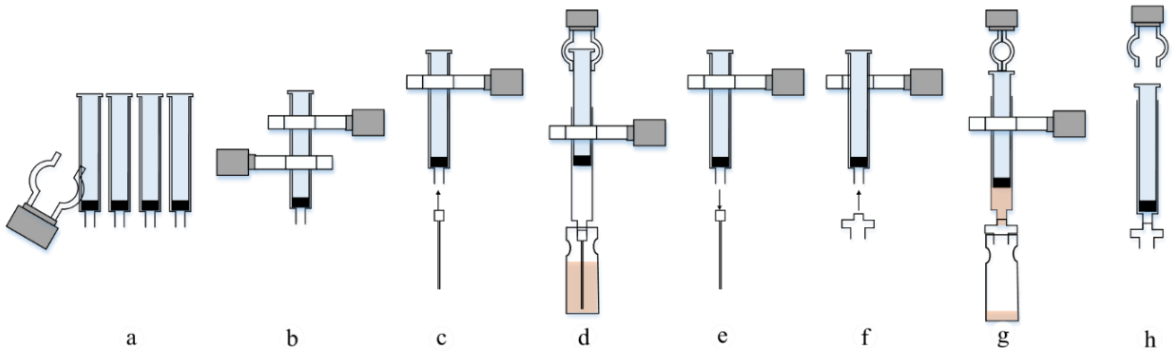


Figure 98: Syringe task (a): pick syringe from rack, (b): transfer to arm, (c): pick cannula, (d): aspirate liquid, (e): reject cannula, (f): pick filter, (g): dispense liquid, (h): reject syringe and filter

Figure 99a depicts all motion elements required for this task in RS-1. The task started at the standby position and the robot right arm moved to prepare position to pick up a syringe (1-2). The syringe was picked up (2-3) and transferred to the left arm (3-4). Next, the cannula is loaded (4-5). A separate motion element positions the right arm to node 5 to actuate the plunger.

The liquid is then aspirated (5-6). The cannula is dropped to a waste bin (6-7) and the robot arm moves to node 8 after loading the filter. The liquid is then dispensed (8-9) through the filter. The syringe is discarded and the arm moves back to the standby position. It can be seen from the Figure 99a that all the motion elements are connected serially and cannot be combined any other way as the node points do not allow so.

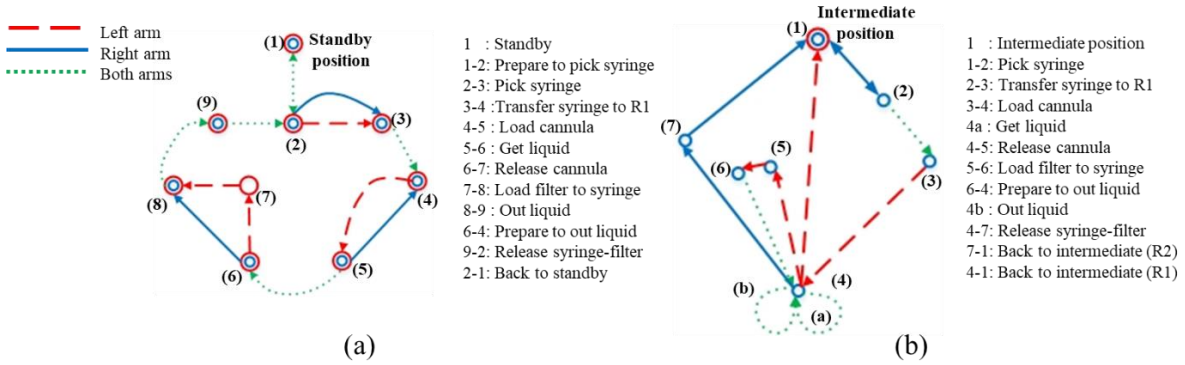


Figure 99: Motion elements for the syringe task; (a): according to RS.1 [111] *(b): according to RS-2*

The task in RS-2 by default starts in the intermediate position (1). The motion to adjust posture to pick syringe is eliminated as the arm is already in the prepare position. The right arm (R2) picks up the syringe during motion (1-2). The robot is not required to place the syringe back; hence, this is a one-way motion. The syringe is then transferred to the left arm between motion (2-3). The left arm reaches node point 3 directly from the intermediate position making the motion twice as faster than in RS-1. The left arm picks up the cannula from one of the ALPs during motion (3-4). The position of cannulas is unreachable with the right arm. At node point (4), the left arm holds a syringe fitted with a cannula and the right arm is in prepare position to actuate the syringe. The motions (4a) depict the aspiration of liquid from a vial after which the arm returns to node 4. The left arm holding the syringe discards the cannula in a waste box placed on the left side of the enclosure (4-5). With addition of an SPE instrument to the workbench, the position of the cannula waste box was changed and the motion element was redesigned accordingly. The arm R1 then picks up a filter from the rack placed on the ALPs in motion (5-6). The dual arms travel back to node (4) for dispensing liquid in motion (4b). As aspiration and dispensing motions start at the same point, they can be done consecutively which might be useful in pre-wetting the syringe if required for future applications. The right arm again assists the left arm to actuate the plunger and filter the solution in a new vial. The syringe and the filter are then transferred to the right arm in motion (4-7) where they are discarded to the waste bucket placed on the right side of the robot. Both the arms move back to the intermediate position as seen in motions (4-1) and (7-1). All the motions in this task are unidirectional, as the tasks require a variety of non-interchangeable labware. However, the number of nodes has been reduced and few unnecessary motions have been eliminated in RS-2. The common node points produce the possibility of changing the motion sequence to adapt the task for new applications.

10.7 Glass Syringe

The glass syringe is an important labware for precise dosing of a liquid stored in septum protected vials. The septum needs to be pierced to aspirate liquid. This is not possible with normal manual pipettes. The plastic tips of a manual pipette are also not conducive to the toxic

reagents. Hence, a glass syringe with needle is used for this purpose (Figure 100). This task has been newly introduced in the RS-2. Both the arms start at the intermediate node (1) as can be seen in the figure. The right arm is repositioned to pick up the syringe during motion (1-2). A motion element from the glass pipette task is reused for this purpose. The right arm picks up the syringe from the temporary pipette stand placed on the workbench following the motion (2-3). The left arm is realigned to reach node point (1-3) such that it is positioned directly above the right arm but not touching it. The liquid source is placed on the cooling plate connected to a cooling pump to keep the derivatizing agent at an appropriate temperature. Both the arms travel synchronously to reach the cooling plate. The arms are position directly above the cooling plate (3-4) and the left arm is positioned such that the finger is in contact with the plunger. The plunger is pulled up by the left arm to aspirate liquid and pushed down to dispense liquid. The glass syringe is washed twice with n-Hexane (4c) and returns to node 4. The reagent is then aspirated (4a). The motion element (4c) for dispensing liquid in the destination vial also starts and ends at node point 4. The motion elements are planned for two different volumes of glass syringes. Both the arms return to the node 3 and the left is not in contact with the syringe plunger any more. The left arm returns to the intermediate position (3-1) at the end of the task. The right arm (R2) replaces the syringe to the temporary rack (3-2) and returns to the intermediate position (2-1).

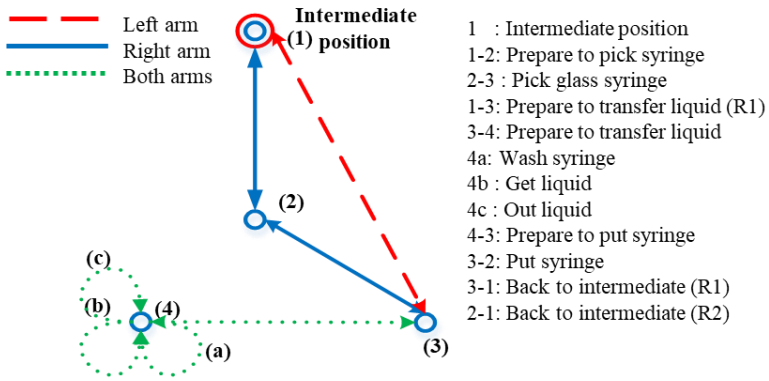


Figure 100: Motion elements for liquid transfer using glass syringe in RS-2

The motion planning of the syringe is important as the syringes as well as the reagents are expensive, the syringe is very fragile and easily broken. As the reagent is used in very small quantities, accurate measurement is also of importance, hence, to avoid any unnecessary damage to labware and to avoid air bubbles in the syringe, the motion of transferring liquid using a syringe has to be carried out very slowly and precisely. After dispensing the liquid, the left arm fingers move away from the plunger while both arms travel back to the cooling rack to avoid finger breakage due to misalignment during the curved motion.

10.8 Vial Task

Most of the sample preparation processes require various types of vials to store reagents or finished samples. The robot performs number of vial actions, most importantly opening and closing of the vial lids as required and transferring the vial from one place to another. The opening and closing of vial is a crucial task, as it requires a high level of accuracy. Figure 101 graphically depicts the vial opening. Vial closing is also carried out using the same motion

elements in a different sequence. The robot manipulates vials as small as of 1.5 mL volume. It is important that the robot gripper fingers be exactly aligned with the vial neck or the lid rack positions. A slight change in position can lead to a failure of the opening and closing task. This may lead to larger and more serious problems in the consecutive robot tasks.

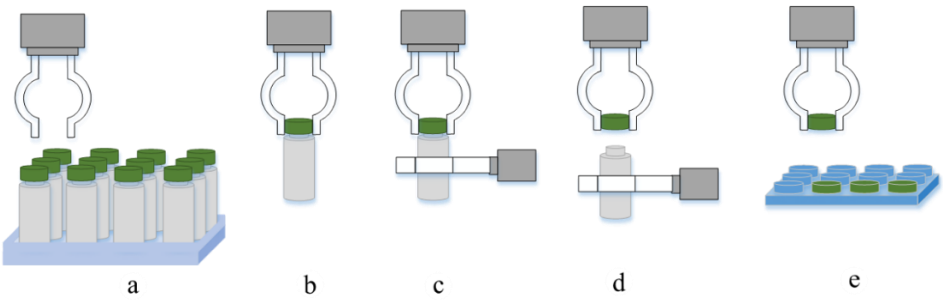


Figure 101: Vial opening (a): approach vial, (b): pick vial, (c): transfer vial to other arm, (d): unlid vial, (e): place vial lid

The task in RS-1 started at the standby position (Figure 102a) and the arm moved to a prepare position to pick a vial (1-2). The robot picked a vial (2-3) which was then transferred to the left arm and the lid was screwed open during motion (3-4) The arm was repositioned to put vial on a lid rack (4-5), the lid was placed on the lid rack (5-6). The vial was then transferred to the right hand and the arm was repositioned to prepare it to place vial on rack (6-3). The vial was then placed on the rack (3-2) and the arm moved back to the standby position (2-1). The start and end positions of all motion elements are different. Although, it stops the robot from performing a wrong motion, the motion elements cannot be combined in any other way to create a new task if required in the future. Change in one end of a motion affects the continuity of the entire task. The task had a number of motions for changing the arm position to prepare it to put or pick a lid or a vial. The motion sequence was very long and tedious during manual robot manipulation.

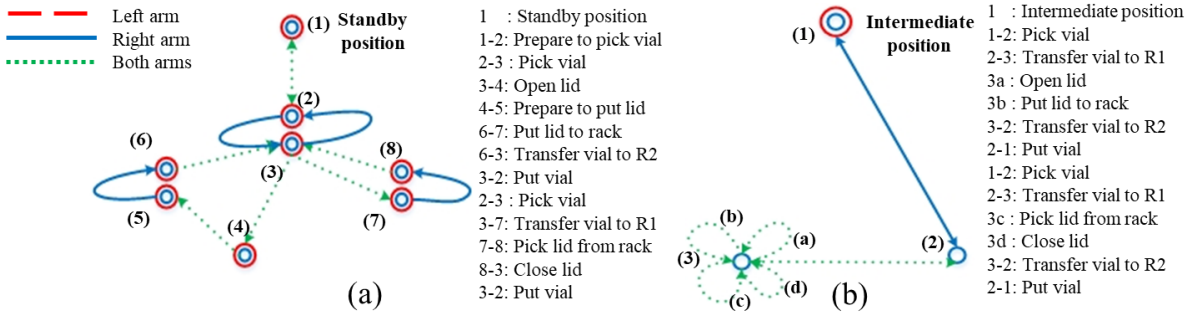


Figure 102: Motion elements for vial task; (a): according to RS.1 [111] *(b): according to RS-2*

The motion elements were changed and the task was re-fragmented so that there are only three node points (Figure 102b). Keeping in mind the expansion of the system, the motions can be mixed and matched to create new tasks. The vial task in RS-2 starts in the intermediate position (1). In order to reach each alp effectively, the torso of the robot is adjusted at an angle during motion and the vial is picked (1-2). The motion element for preparing to pick seen in RS-1 has been eliminated. The key point (2) is floating without fixed coordinates. The vial is transferred to the left arm (R1) during motion (2-3) so that it is grasped tightly between two fingers. The right arm then opens the vial (3a) and places the vial lid to the vial rack (3b) placed on another ALP. The open vial is transferred to the right arm (3-2), which places it back in the

vial rack (2-1). The same sequence of motion elements is followed to close the vials. The open vials are transferred to the left arm. T node 3, the robot can perform motion elements to pick lid from rack (3c) and close the vial (3d). The vial is transferred back to the right arm to be placed in the vial rack. In RS-1 the motion to transfer vials from one arm to the other was integrated in the open vial and close vial motion elements. Instead, these motions are independent in RS-2. The vials can be transferred from one arm to another without performing any other operation on them. This function adds more possibilities for future adaptations and task variations.

The vials have varied requirements due to their sizes. The 2 mL, 4 mL, 10 mL and 22 mL vials are placed in a 4x3 configurations. The larger 40 mL vials are placed in a 3x2 configurations and are newly introduced in RS-2. Similarly, a vial block specially designed for SPE applications containing 2 mL vials in the 6x4 configuration has also been newly implemented in RS-2. Hence, the vial motion elements were adapted for handling the 40mL vials and the SPE vial block. Due to difference in configuration, the calculations required for manipulating the various vials are unique. The configuration to be implemented is dependent on the vial type and is hard coded in the motion element program.

The motion element is also adapted to pick up 1.5 mL vials without lid. The neck diameter of these vials is very narrow (6 mm), which at times leads to the gripper fingers not identifying their presence. Hence, the motion element is created such that the vial is picked from the body instead of the neck. All the other vials are grasped at the neck, as the design limitation of the gripper finger and vial sizes do not allow enough space between consecutive vials to be grasped by the body.

10.9 Ultrasonic Shaker and Shaker Task

Sample solutions and reagents are regularly transferred to the shaker or the ultrasonic bath for efficient mixing of the reagents. The ultrasonic bath only supports 2 mL vial formats. Either one or both of the methods may be implemented to achieve a homogenous mixture. It is important that the robot is capable of moving the vials between the ALPs, shaker and ultrasonic bath in any order required. Figure 103 graphically depicts the motion elements for the ultrasonic shaker or shaker tasks. The robot in intermediate position picks up the vials and directly places them on the shaker or in the ultrasonic shaker bath. The application considered in this dissertation, requires the vials to be transferred from the workbench to the shaker, followed by the ultrasonic shaker. The vials were then brought back to the shaker from where they were transferred to the workbench. This sequence of motion is considered as one task for the purpose of this dissertation. However, the motion elements can also be performed such that the shaker or ultrasonic bath are used independently.

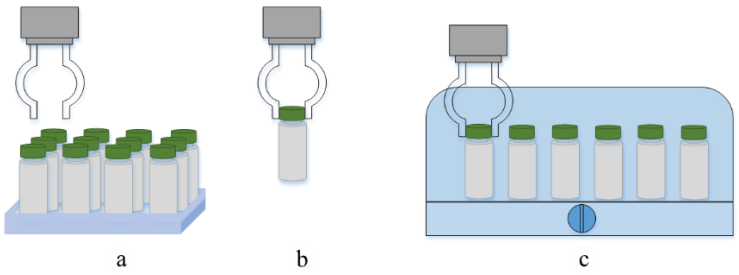


Figure 103: Ultrasonic bath and shaker task: (a) pick vial, (b) transfer vial, (c) place vial in ultrasonic bath / shaker instrument

The RS-1 (Figure 104a) required the right arm in standby position to move to a prepare position in order to pick up a vial (1-2). The vials were picked from the shaker (2-3). A new motion element adjusted the robot posture to access the ultrasonic bath (3-4) and vials were placed in the ultrasonic bath (4-5). The robot went back to a prepare posture at node 2 to pick up the next vial. The vials were transferred individually to the shaker. The timer on the ultrasonic bath was set. The motion 2-5 positioned the arm to access vials in the ultrasonic bath, which were picked and put to the shaker using three different motion elements in a sequence (5-4, 4-3, 3-2). The motion elements all had a different start and end point. Hence, the motion elements have to be carried out in the same sequence in order to complete the task successfully.

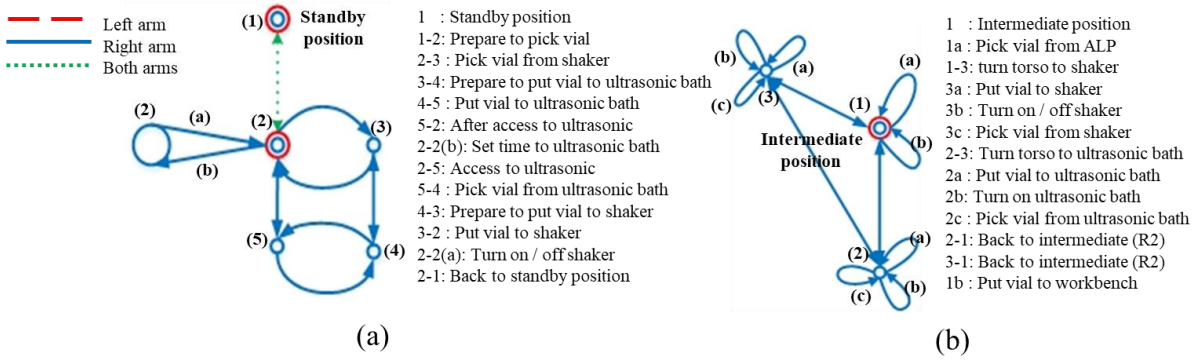


Figure 104: Figure depicting the motion elements for mixing task with shaker and ultrasonic bath; (a): according to RS.1 [111] (b): according to RS-2

However, RS-1 did not allow the flexibility to use the shaker and ultrasonic bath in a reverse order, or to use the ultrasonic bath independently. Vials could not be transferred between the workbench and the ultrasonic bath. Hence, the motion elements were changed to make the task flexible. All the motion elements required for using the ultrasonic bath and the shaker start and end at the intermediate position (Figure 104b). Only the right arm is used for these tasks. The right arm in the intermediate position picks up the vial from the workbench (1a). During the next motion, the robot torso is rotated to reach the shaker places the vial in the shaker and comes back to node 1. All the vials are placed and the shaker is turned on by pressing the button. Contrary to RS-1, there is no need to create a new motion element to adjust the robot posture for placing vials in shaker. At the end of shaking, the vials are picked up from the shaker and robot comes back to node 1. The torso is rotated again to orient itself to the ultrasonic bath and the vials are placed in the ultrasonic bath one by one. As all the motions of this task start and end at one point, the motions can be implemented in any order to create new combinations for future applications. The shaker or the ultrasonic bath can be implemented independently in RS-2. The shaker is turned on or off with the press of a button by the robot fingers. The ultrasonic bath is also turned on by the robot with the help of a turning switch. The major challenges in these tasks are the sustained correct positioning of the vials during transfer. If the vials are not held exactly straight, there might be a collision with the instrument. It is especially important in the ultrasonic bath task the bath has a specially designed 3D printed plate to hold the vials in place. It is the farthest instrument placed on the right side of the robot. In this position, the robot arm is nearing its limit of motion. The arm in this position does not travel exactly straight which introduces a minor offset in the vial positions at destination. Care has to be taken to accurately adjust this offset while programming the motion element.

10.10 Liquid Chromatography

Figure 105 depicts the task of feeding finished samples to the LC autosampler for analysis. The samples can be contained in either an MTP or 1.5 mL vials. The samples are transferred to an LC tray on the workbench. The LC door is opened with the help of one arm and the tray is placed in the LC autosampler. The door of the LC is closed by the robot. The automated measurement system is started manually by a laboratory technician.

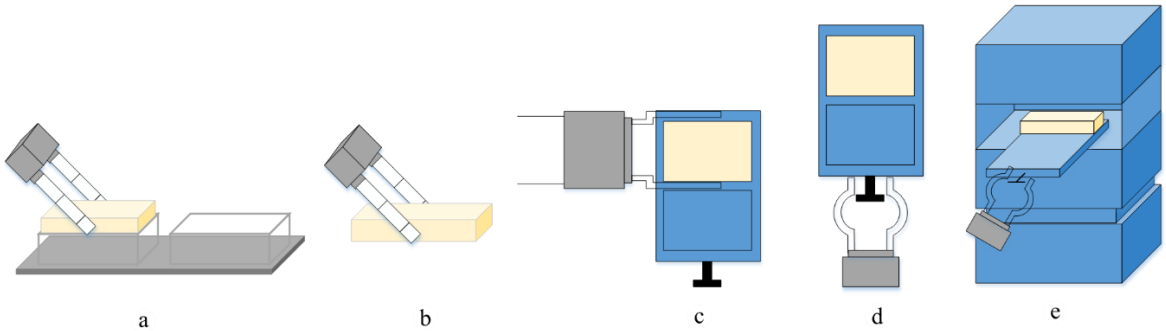


Figure 105: LC task (a) pick labware with arm 1, (b) transfer labware, (c) place labware on tray with arm 1, (d) transfer tray with arm 2, (e) place tray in LC with arm 2

The RS-1 requires the arms to be in standby position at the beginning. The left arm moves to a preparatory position and transfers the labware to the LC tray. The right arm opens the LC

door with the press of a button. The right arm transfers the tray to the LC and closes the LC door. The right arm then goes back to the standby position. At the end of the measurement, the right arm opens the LC door, transfers the tray to the workbench and closes the LC door. The left arm is programmed to place the labware into a hotel. As seen in other tasks the motion elements can be combined in one unique way in order to perform the LC task. The robot motions have been changed in RS-2 such that the operations of both the arms are independent of each other.

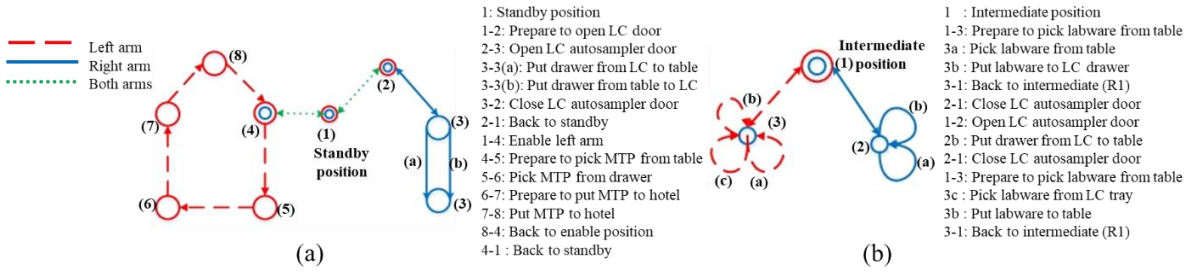


Figure 106: Motion elements for LC-autosampler tasks; (a): according to RS.1 [111] *(b): according to RS-2*

Both arms start in the intermediate position (1) in the RS-2. The LC tray has a special position on the workbench so that it can be easily picked up by the robot. The left arm is moved to the workbench key point to pick the labware from the table during motion (1-3). The arm picks up the samples (3a), transfers it to the LC tray (3b) and returns to the intermediate position (3-1). The right arm starts in the intermediate position, opens the LC door and the motion ends at node point 2. From this node point, the robot transfers the LC tray to the LC autosampler. After closing the door, the robot goes back to initial position via motion (2-1). At the end of the measurement, the right arm opens the LC, transfers the LC tray to the workbench and closes the LC door. The right arm then goes back to the intermediate position (2-1). The left arm then picks up the labware from the LC tray (3c). The labware can be transferred to the workbench, hotels or the shelves as required. The common node points provide the flexibility to change the destination of the labware as required with minimum effort. Multiple motion elements have been combined into one in RS-2 simplifying the task and shortening the task job.

10.11 Gas Chromatography

The gas chromatography mass spectrometer is used for analysis of volatile compounds. The GCMS is placed to the left of the robot. The GC autosampler is a circular disc with 100 radially positioned slots to hold 2 mL sample vials. The positioning of the vial slots makes it difficult to create a robot motion element with a specific algorithm for feeding the samples. Hence, all accessible sample slots have to be taught to the robot individually. Figure 107 graphically depicts the steps involved in transferring the sample vials to a GC autosampler.

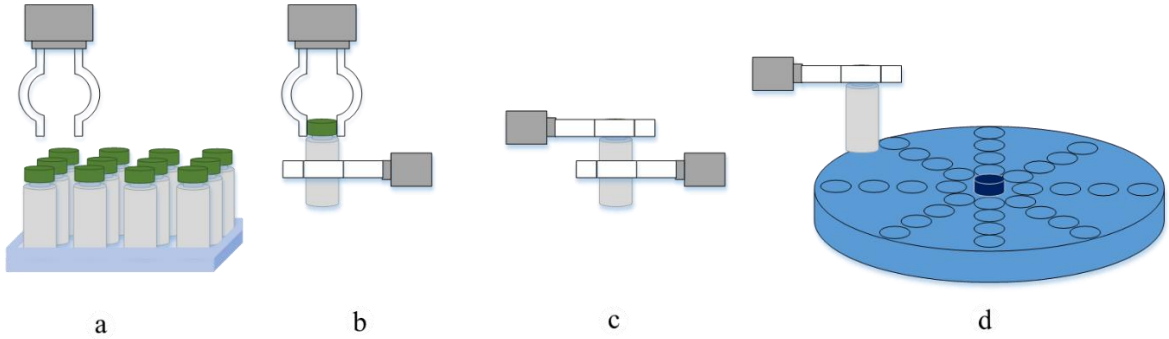


Figure 107: GC task (a): pick vial, (b) transfer vial to arm 1, (c) transfer vial to arm 2, (d)place vial to GC autosampler

The motion elements in RS-1 were programmed to only access the first 15 wells of the GC autosampler (Figure 108). The motion elements for GC task were changed in RS-2 to be able to combine with the new vial motions.

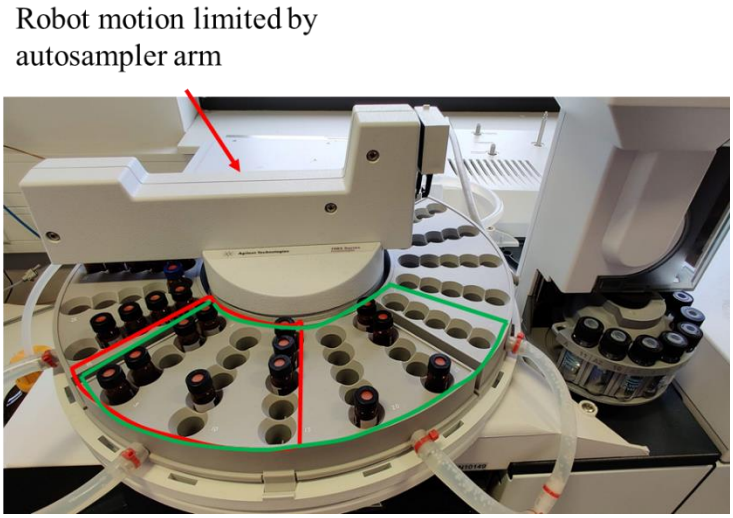


Figure 108: Wells marked by red programmed for RS-1, wells marked by green programmed for RS-2

The motion of the right arm was also edited to reach the destination in fewer steps. In order to maximize the number of samples measured, additional wells are programmed such that the robot can now access well numbers 1-30 on the auto-sampler. The well access is limited by the reachability of the robot arm. Furthermore, the rotating arm on the GC auto-sampler also obstructs further access of wells by the robot. Figure 109a depicts the motion element relationship for the GC task in the RS-1.

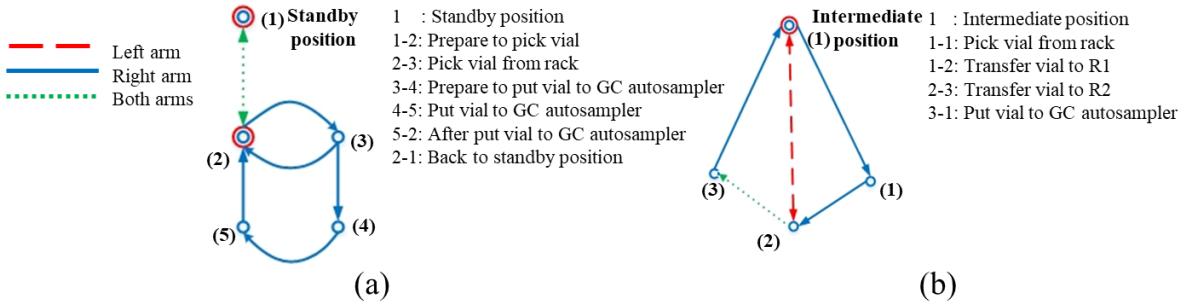


Figure 109: Motion elements for the task of loading sample vials to the GC autosampler, (a): according to RS.1 [111] *(b): according to RS-2*

The GC task in RS-2 initiates in the intermediate position (Figure 109 b). Right arm (R2) travels to the floating key point (2) to pick up a vial from the workbench depicted by (2a). The dual-arm robot can only reach the GC in one particular posture. Hence, the vial in the right arm is transferred to the left arm in motion (2-3). The posture of R2 is changed to suit the GC positioning during motion and the vial is transferred back to the right arm (3a). The vial in the right arm is then placed in the accurate GC slot in motion (3-4). The arm returns to the initial position in motion (4-1). All the motions in this task can be retraced to the previous step. For future expansion, the same motions can also be used for unloading vials from the GC and placing them on the workbench.

10.12 Thermoshaker

The thermoshaker enables the efficient homogenization of reagents at an elevated temperature. The robot removes the shaker lid. The labware is picked up from the workbench and transferred to the thermoshaker. The lid of the shaker is then replaced before starting the shaking motion. It is placed on the workbench on the left of the robot and the various steps are shown graphically in Figure 110.

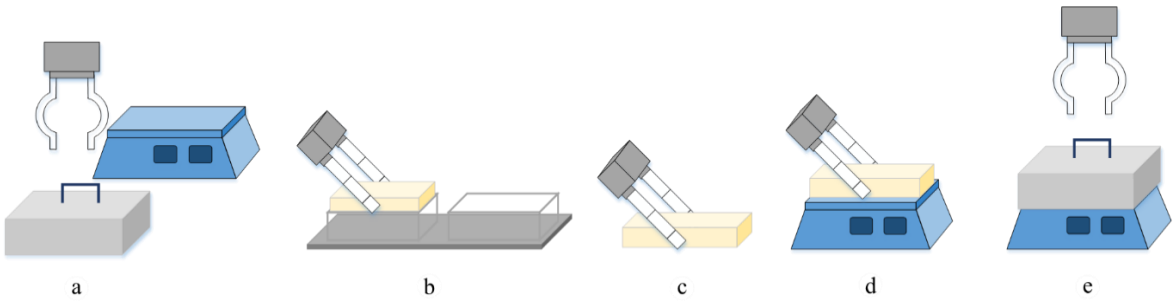


Figure 110: Thermoshaker task (a) open thermoshaker with arm 2, (b)pick labware from workbench with arm 1, (c): transfer labware, (d) place labware to thermoshaker, (e) close thermoshaker

The motion elements of the thermoshaker include opening/closing the thermoshaker with the right arm and transporting the labware with the left arm. In RS-1 (Figure 111a), the task was very long and time consuming. The task started in the standby position. The robot rotated 180 degrees around its axis (1-2) and required an elaborate arm adjustment to reach the instrument from over the body with its right arm to open it. Two motion elements were used for this purpose, motion 2-3 and 3-3a. The robot then came back to the standby position using two motion elements (3-2 and 2-1). The left arm moved to the ‘prepare to pick labware’ position (motions 1-4 and 4-5). The MTP, vial rack as well as the MTP lid was transferred to

the instrument from the workbench. The left arm then went back to the standby position (motions 5-4 and 4-1). The robot again rotated 180 deg around the torso to close the lid with the right arm. The fingers pressed the start button on the thermoshaker to start the machine. The motion was very slow and crude for an observer. Hence the motions were re-planned (Figure 111b) and made faster and simpler in RS-2.

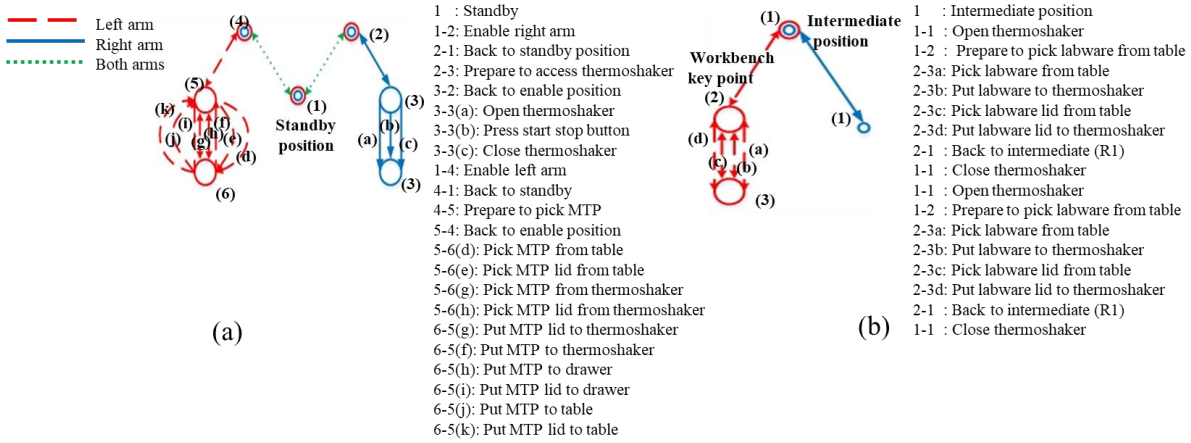


Figure 111: Motion elements for thermoshaker task; (a) according to RS.1 [111] *(b): according to RS-2*

The RS-2 involves the robot initiating the task in the intermediate node (1) and turning 45 degrees on its left in a much shorter motion compared to RS-1 to reach the thermoshaker (1-1). This motion requires a third of the time as required in RS-1. The right arm opens the lid of the thermoshaker (1-1), places the lid on the workbench and returns to intermediate node in one motion. The left arm adjusts position (1-2) to workbench key point to pick up labware from the workbench. The arm travels to floating node (3) to pick up the labware or the labware lid from the appropriate ALPs and travels back to workbench key point (2). The labware is transferred to the thermoshaker and returns to the workbench key point. The left arm travels to the intermediate position in motion (3-1) after labware transfer. The right arm then places the thermoshaker lid back on following motion (1-1). The thermoshaker is turned on and off directly from the controlling PC vial the USB connection. Often volatile reagents like methanol and hexane are used to prepare the samples. At elevated temperatures, these reagents are evaporated faster. Hence, it is important that the labware be appropriately lidded.

10.13 Solid Phase Extraction

The SPE module has been newly integrated in the RS-2 (Figure 112). The module is placed on the left corner of the workbench such that the cartridges and the glass vials can be loaded onto the carriage of the SPE. The SPE is actuated directly by the SAMI Ex. The cartridge racks are heavy and are stored in the overhead shelves. The rinsing blocks and 2 mL vial racks specifically designed for the SPE are stored in the hotels.

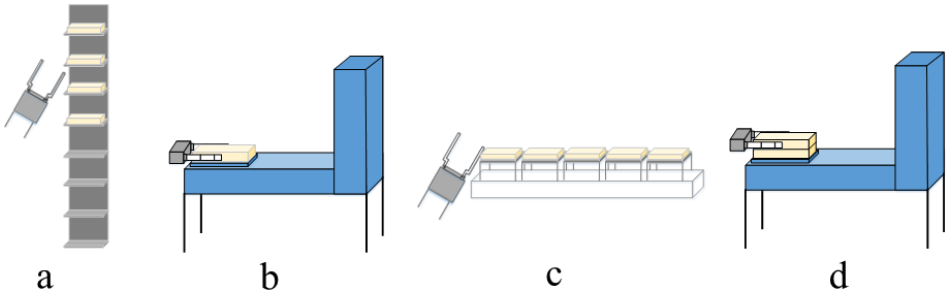


Figure 112: SPE task (a): pick rinsing block from hotel, (b) place labware on SPE, (c) pick cartridge from shelf, (d) stack cartridge on previous labware

Figure 113 depicts the motion diagram for the task.

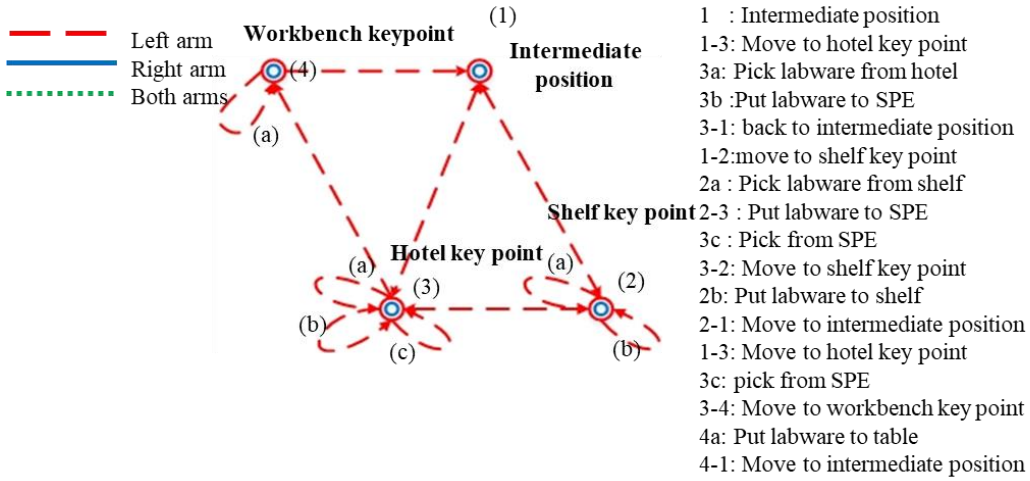


Figure 113: Motion elements for SPE task in RS-2

Starting from the intermediate position (1), the arm reaches the hotel node (3). The rinsing block placed in the hotel is picked up and the arm goes back to the hotel key point (3a). The labware is placed on the SPE (3a). This motion ends in the hotel key point. In order to retrieve the cartridges placed in the overhead shelves, the robot moves first to the intermediate position and then to the shelf key point (1-2) where the labware is picked up (2a). The arm moves back to the hotel key point to place the cartridges on top of the rinsing block (2-3). The arm returns to the intermediate position. The liquids are pipetted in the SPE cartridges following the motions explained in the pipette task earlier in this chapter. The SPE carriage is actuated directly by SAMI Ex and positive pressure is applied. In order to collect the target liquid, the cartridges are placed on the shelf and the rinsing block is put back to the hotel. At the end of this motion, the robot arm is at the hotel key point. The arm retrieves the vial adaptor from the hotel (3a) places it on the SPE (3b). The cartridges are brought down from the shelf and placed on top of the vial adaptor (3b). The arm moves to the intermediate position (3-1). After the actuation of the SPE, the arm starting in the intermediate node, puts back the cartridges to the shelf. The arm picks labware from the SPE, and transfers it to workbench key point (3-4) and places it on the designated ALP (4a) for further processing. The labware travels directly from each of the hotel and shelf key points to the SPE module without the need of changing the grip angle on the regrip-station.

11 Realized Applications

Two automated sample preparation applications are designed in order to test and validate the robotic system.

11.1 Application I: Sample Preparation for Detection of Chiral Enantiomers

Certain molecules depict the property of chirality, where the two enantiomers are mirror images of each other. Naturally occurring organic compounds such as proteins, carbohydrates, enzymes, hormones are found to be chiral. Amino acids have naturally occurring L and D isomers. Bothe these isomers play important independent roles in bodily functions. Chiral enantiomers have been seen to have different pharmacological and toxic behaviors. Their effect and toxicity on the human body depend upon their reaction with the receptor cells. In case of a racemic mixture, one enantiomer might be beneficial to the body and the other not. This might lead to unexpected pharmacological behavior and harmful side effects [119]. Thus, pharmaceuticals should only be used in their pure enantiomeric forms. Enantioselective synthesis and separation are hence currently important scientific fields [20],[32],[97],[98].

The structural information of amino acids is usually obtained by conventional methods such as liquid chromatography mass spectrometry (LC/MS) and gas chromatography mass spectrometry (GC/MS). These methods are relatively time consuming along with high costs incurred for columns, additives, and reagents. Fast and reliable analytical methods are required for processing an increasing number of samples, especially in drug development and pharmceutical quality control. A method based on parallel kinetic resolution and mass spectrometry with pseudo-enantiomeric mass-tagged auxiliaries has been developed that enables enantiomeric excess determination of proline in approximately 2 min per sample [121]–[123]. Differentiation of the enantiomers is possible due to differences in their reaction kinetics with other chiral compounds. The chiral mixture of the two enantiomers of the amino acid proline is derivatized with two auxiliary compounds. The mass difference of the auxiliaries causes four reaction products with two distinct masses. The ratio of these masses can be used to calculate the enantiomeric excess.

With this method, we attempt to measure the amount of each chiral enantiomer in each sample solutions. The robot dilutes the proline enantiomers to the required concentrations. These solutions are mixed together in pre-decided amounts to create standard solutions with known concentrations. These solutions are then analyzed to create a calibration for the subsequent measurement of sample solutions with an unknown enantiomeric excess. In this work, instead of unknown samples, standard samples with an enantiomeric excess of -100 ee%, -50 ee%, 0 ee%, 50 ee%, 100 ee% were used. The measured data is analyzed for comparison of the output to the expected results. This is a method validation application for evaluating the precision and recover rate achieved using the robotic system

11.1.1 Description of Process

The dual-arm robot is used for the automated sample preparation for determining the enantiomeric excess in a chemical mixture. Figure 114 provides an overview of the sample preparation process in the form of a block diagram.

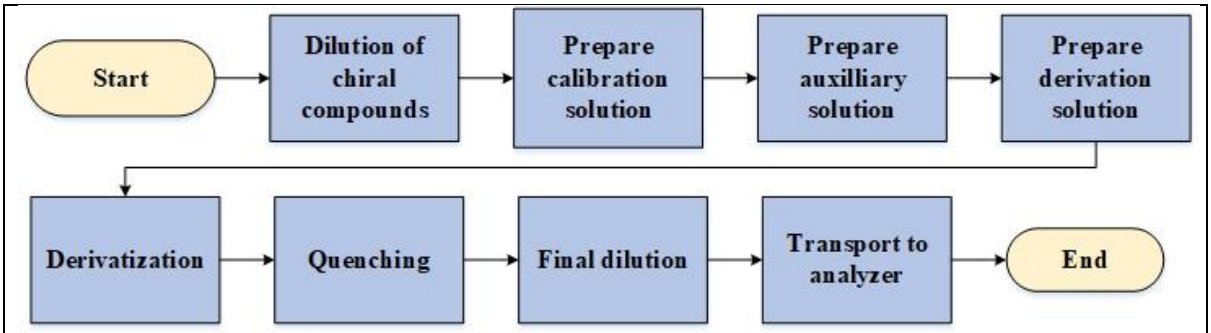


Figure 114: Block diagram depicting the steps of sample preparation for detecting chiral enantiomers

In this automated process, the starting material (source powders and solutions) are prepared manually. The amino acids L-proline and D-proline are weighed and pipetted in two separate 10 mL vials. The derivatizing chiral agents L-FDVA (Nα-(2, 4-Dinitro-5-fluorophenyl)-L-valinamide) and D-FDLA (Nα-(5-fluoro-2, 4-dinitrophenyl) D-leucinamide) in powder form are weighed by a technician and stored in two separate 10 mL vials. Water for the dilution of proline solution is also filled in 22 mL vials. The remaining source liquids are stored in a multi-compartment reservoir. The first container of the reservoir is left clean and empty for later use. The second to fifth containers are filled with methanol, the sixth container is filled with hydrochloric acid (HCl), and the seventh container is filled with Ammonium bicarbonate (NH_4HCO_3). The last compartment of the reservoir is filled with acetone. All these source materials are provided to the robotic system to perform the actual sample preparation task. The automated sample preparation process is depicted in Figure 115.

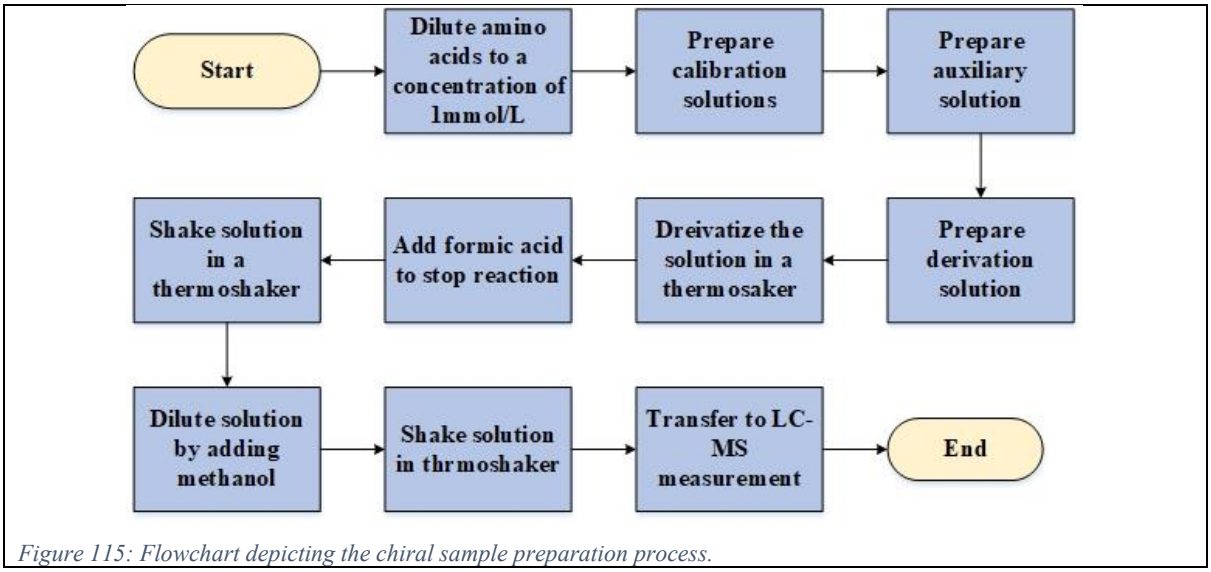


Figure 115: Flowchart depicting the chiral sample preparation process.

The automated task is carried out as follows:

Diluting chiral compounds: In this process the amino acids L-Proline and D-Proline with molar concentrations of 50 mmol/L are diluted to 1 mmol/L. Each amino acid is pipetted (120 µL) separately into two glass vials containing 5,880 µL water each. The 200 µL pipette is used for this purpose. Finally, one vials contains a diluted solution of L-Proline and the other contains diluted D-proline.

Calibration solutions: Before preparing the samples to be tested, a calibration is carried out in order to establish the levels of concentration for reference. The diluted L-Proline and D-Proline is distributed in five different empty vials in varying ratios; (L: D): 2000 µL:0 µL (100% ee), 1500 µL:500 µL (50% ee), 1,000 µL:1,000 µL (0% ee), 500 µL:1500 µL (-50% ee), 0 µL:2000 µL (-100% ee). All the vials will contain 2 mL liquid at the end of this step. The enantiomeric excess of the test samples can be found out from the calibration curve produced by these solutions. The 1,000 µL pipette with volume set at 500 µL is used for the liquid transfer.

Auxiliary solution: The auxiliary solution is a derivatizing agent for determination of the chiral enantiomers. At the beginning of the process, 7.50 mg L-FDVA powder and 7.85 mg D-FDLA powders are weighed manually and stored in 10 mL vials. These powders are used for the auxiliary solutions. The robot pipettes 2 mL acetone in L-FDVA powder. The resulting solution is pipetted into the vial containing D-FDLA powder. This step is carried out four times. The resultant solution is an 8 mL auxiliary solution containing a mixture of acetone and L-FDVA and D-FDLA powders. The solution is later pipetted in a MTP using a multi-channel pipette. For this the auxiliary solution is transferred from the vial into the first container of the reservoir (which was left empty in the manual step).

Derivation solution: All the solutions generated until now are transferred to a 96 well MTP where the chiral derivatives will be produced. The first 15 wells, serially from A1 to G2, are used for calibration. 50 µL of the 100% ee solution is pipetted in A1, B1, C1 wells; 50% ee solution is D1, E1, F1 wells; 0% ee solution in G1, H1, A2 wells; -50% ee solution in B2, C2, D2 wells and -100% ee solution in E2, F2, and G2 wells. The well H2 is filled with water. The 100 µL pipette is used for this purpose. The remaining wells are filled with 50 µL of the diluted L-proline or D-proline standard solutions. For the purpose of this study, the 3rd and 4th column of the MTP was filled with the 100% ee solution, 5th and 6th with 50% ee solution, 7th and 8th with 0% ee solution, 9th and 10th with -50% ee solution and finally 11th and 12th with -100% ee solution. These samples were treated as unknown samples. Next, 100 µL auxiliary solution, using the 100 µL multi-channel pipette, is pipetted into each of the wells. Lastly 20 µL NH_4HCO_3 solution, using 10 µL multichannel pipette, is pipetted into each of the wells. The NH_4HCO_3 promotes reaction between the auxiliary solution and the amino acids.

Derivatization: The MTP is transferred to the thermoshaker. Derivatization takes place at 750 rpm for 20 minutes at room temperature.

Quenching: The MTP is transferred from the thermoshaker back onto the worktable. 10 µL HCl is pipetted into each of the wells of the MTP. The 10 µL multi-channel pipette is used for this. The HCl stops the derivatization reaction. The plate is transferred back to the thermoshaker for 2 minutes for thorough mixing of the liquids.

Final Dilution: The liquid from the MTP is diluted to a suitable concentration for analysis with mass spectrometry. A new empty MTP is used. 450 µL methanol is pipetted into each of the well of the new plate. 50 µL sample solution is transferred from the first MTP into the corresponding well on the second MTP. The ratio between sample solutions and methanol can be changed to obtain final solution of varying concentrations. The plate is transferred onto the thermoshaker for 2 minutes for mixing at room temperature.

Transport: The plate is finally transferred onto the mass spectrometer robotically for analysis.

11.1.2 Results and Discussion

It was observed that the measurement results of the robotically prepared samples in RS-2 are comparable to the manual samples as seen in Table 13. The table shows the ratio of the masses of the enantiomers present in each of the samples. The compounds formed by the amino acids with the two auxiliary compounds have the masses of m/z= 394.14 and m/z=408.16 and are indicated as Int M1 and Int M2 in the result Table 13 [121]–[123].

Table 13: LC Measured values of both enantiomers in manually prepared and robot prepared samples

	Manual Sample Preparation		**Robotic Sample Preparation (RS-2)**	
ee%-Value	***Int Mass 1 / Int Mass 2***	***Average***	***Int Mass 1 / Int Mass 2***	***Average***
100	1.37	1.33	1.33	1.30
100	1.29		1.14	
100	1.33		1.42	
50	1.17	1.13	1.07	1.10
50	1.15		1.28	
50	1.08		0.95	
0	0.93	0.88	0.78	0.86
0	0.84		0.97	
0	0.87		0.83	
-50	0.61	0.65	0.63	0.60
-50	0.65		0.59	
-50	0.68		0.57	
-100	0.40	0.41	0.44	0.39
-100	0.41		0.42	
-100	0.42		0.30	

It can be seen that the robotically prepared samples are as accurate as the manually prepared samples thus verifying the accuracy of liquid handling and transfers. There is an error margin of ±4% between both the measurements (considering manual sample values as 100%).

Figure 116 below depicts the graphical representations of the measured enantiomeric excess against the intensity ratio of the enantiomers.

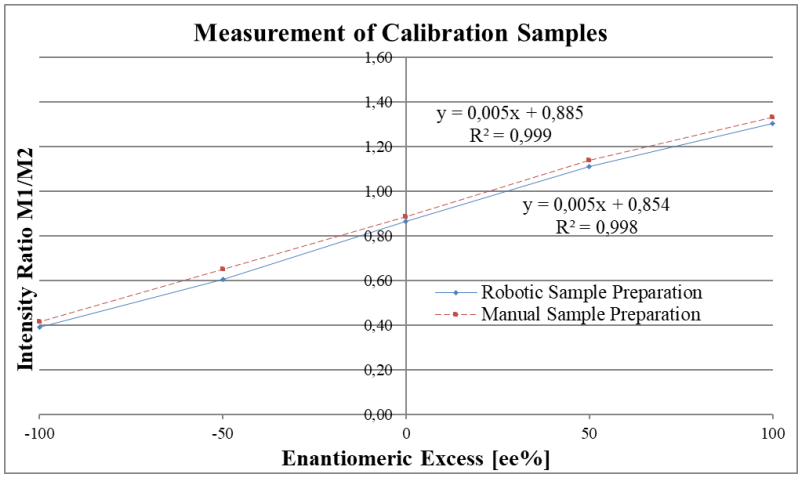


Figure 116: Graphical comparison of the calibration measurements prepared manually and using the dual-arm robot

It can be seen that the graphs are linear and nearly identical in the manual as well as the robotic sample preparation processes. The coefficient of determination in the manual process is 0.999. The robotic process is comparable with a coefficient of determination of 0.998. Thus, there is no loss of efficiency with the automated process and the system is reliable. Besides calibration, eighty remaining wells of the MTP are filled with the prepared standard samples and treated as unknown samples in RS-2. Figure 117 depicts the results of analysis of these samples. The average intensity ratios of 16 samples are plotted against the corresponding enantiomeric excesses.

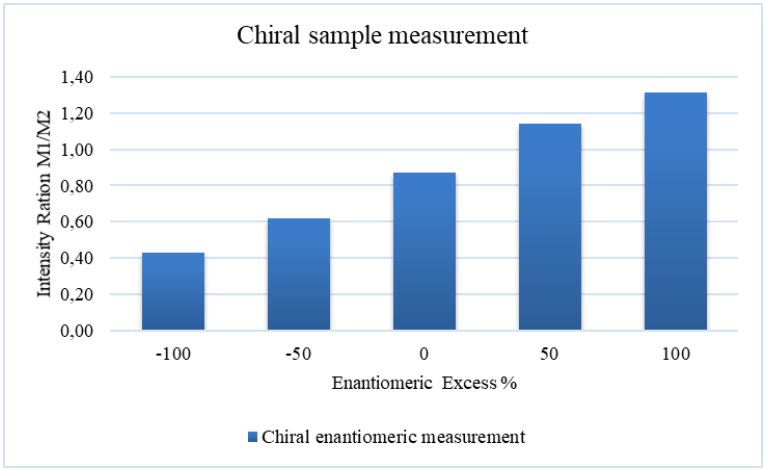


Figure 117: Graph depicting the average intensity ratios of unknown samples:

The time required for individual tasks as also for the total process according to RS-1 and RS-2 was recorded and compared. In RS-1, robot motions for labware transfers between tasks was long, which increased the overall time required for the process. Table 14 summarizes the observations from both the methods in a tabular format. In the table the time is specified in the hh:mm:ss format. Motion planning method I according to RS-1 was the sample preparation system already implemented on the robot before the start of this dissertation. Motion planning method II was the sample preparation process done in RS-2 which followed the exact steps as in method I.

Table 14: Speed comparison between the two automated methods.

Step	Task	Motion Planning Method I	Motion Planning Method II	Motion Planning Method III
		RS-1 (hh:mm:ss)	***RS-2 (hh:mm:ss)***	***RS-2 (hh:mm:ss)***
1	Diluting chiral samples	00:12:59	00:03:26	00:03:26
2	Preparing calibration solution	00:05:57	00:04:50	00:04:50
3	Preparing auxiliary solution	00:15:05	00:10:44	00:03:33
4	Preparing derivatization	00:24:01	00:22:08	00:35:45
5	Final dilution	00:23:03	00:15:40	00:15:40
	Time required for pipetting	**01:21:05**	**00:56:48**	**01:03:14**
	% Time saving between Method I and Method II		**30.3%**	**22%**

A significant time difference is seen in step 1, 3 and 5. The transportation of tip boxes from the shelf to the table are significantly faster than the in method II than in method I. The motions for liquid transfer are also faster. Method II requires 25 minutes less than method I resulting in 30% time saved compared to method I.

In order to improve the process flow and for better measurement results, the sample preparation process was performed again (Motion planning method III). The step 3 in method I and method II utilizes a glass pipette for transferring the auxiliary solution to a tank. This liquid transfer is time consuming, as the pipette used in a previous task needs to be paced back and glass pipettes have to be picked up. The speed of glass pipette task is slower than a manual pipette task as well due to the fragility of glass pipettes. The robot used a 5 mL manual pipette instead of a glass pipette reducing the time by 7 minutes for step 3. Lab technicians also use air displacement pipettes for this liquid transfer step during manual sample preparation. The liquid reservoirs in method I and method II did not have a lid. However, lack of reservoir lids leads to evaporation of volatile reagents. Hence, a reservoir lid and MTP lid was used for keeping the source liquids and samples covering the in method III. The robot needs to lid and unlid the reservoir multiple times in step 4. Hence, the method III is approximately 11 minutes slower than the method I.

Due to replacing the glass pipettes with manual pipettes, however, the method III is faster and requires a total of 63 minutes 14 seconds. Method I requires a total time of 81 minutes 5 seconds. It is observed that there is a 22% time saving with the new approach. A skilled lab technician preparing the calibration samples manually requires 30 minutes 40 seconds for the complete process. The automated process requires twice as long as the manual process. This delay in time is caused by significant times in transferring the required labware to the worktable. Laboratory technicians collect all the required labware and pipettes on the workbench prior to sample preparation. The robot cannot do this, as the workbench only has a limited number of positions. Further, the motions are also restricted by reachability of arms on the workbench. Hence, even if spare ALPs are available, they may not be of practical use for all tasks. The speed of the robot is adjusted to the maximum value appropriate to any particular step / motion.

11.1.3 Batch Sample Preparation

Four batches of the above described process with 12 samples each were run successfully on the RS-2. All the required labware was placed in the overhead shelves and the hotels at the beginning of the process. The MTP and reservoirs are placed in the hotels. The various tip boxes and 22 mL vials were placed on the overhead shelves. The thermoshaker and the mass spectrometer are utilized by all the families. Table 15 details the list of labware required for preparing four batches of samples and the labware positions before/after and during the run.

Table 15: List and quantity of labware for four batches of samples for analysis of chirality of compounds

Labware	Quantity	Storage Position	Position on bench during run
Reservoirs with lids	4	Hotels	ALP 3
MTP with lid	4	Hotels	ALP 4
MTP without lid	4	Hotels	ALP 1
4 mL vial rack containing 5 vials each	4	Hotels	ALP 0
22 mL vial rack containing 5 vials each	4	Shelf	ALP 9
1 mL tip box	4	Shelf	ALP 7
300 µL tip box	4	Shelf	ALP 7
10 µL tip box	4	Shelf	ALP 7
Thermoshaker	1	Thermoshaker position	Thermoshaker position
Mass spectrometer	1	LC Position	LC Position

The four labware families utilize all 16 shelf positions. The tip boxes cannot be placed in hotels due to size incompatibility. Hence, there is no more space for placing labware for more families on the shelves. The robot transfers labware for each family on the workbench (Figure 118) and places it back to their initial positions after the sample preparation. The finished samples ready for measurement were transferred to the LC autosampler. The robot retrieved the measured samples from the previous batch from the LC autosampler before preparing the next batch of samples.

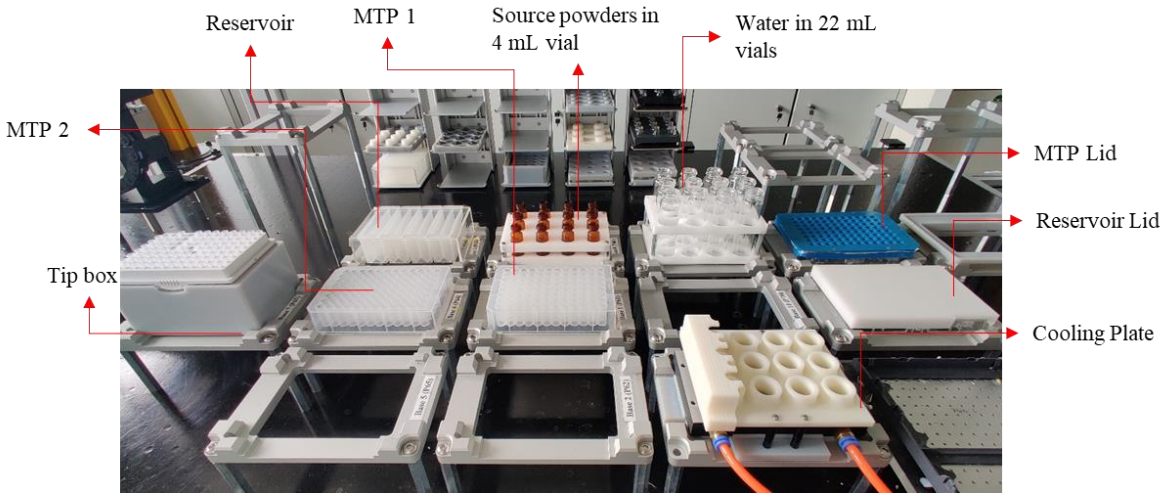


Figure 118: Labware placement on workbench during sample preparation

The important obstacle encountered in batch processing of these samples was that, the robot environment has only one space each to place the 10 mL and 5 mL tips. The tips are larger in

size than the other tips for lower volumes. Placing the 10 mL and 5 mL tips on other ALPs creates an obstacle to the robot arm movement. Thus, in spite of large volume transfers, these tips cannot be used for batch processing with multiple families. Alternatively, 1 mL pipette has be employed that increases the number of pipetting steps and may as well potentially introduce more pipetting errors. Table 16 lists the various motions required to perform the sample preparation process. The table provides the success and failure of each task. The motions were extensively tested before application. A detailed report of the tests and changes made are included in the appendix in detail.

Table 16: List of all the motions required for chiral sample preparation with success / failure for each well column in MTP

Motion Element	Motion Performed
VI_SJ_PICK_VIAL.JBI	Pick vial from table
VI_SJ_TRAN_VIAL_T_R1.JBI	Transfer vial to R1
VI_SJ_OPEN_LID_T_VIAL.JBI	Open lid
VI_SJ_PUT_LID_T_RACK.JBI	Put lid to table
VI_SJ_PICK_LID_F_RACK.JBI	Pick lid from table
VI_SJ_CLOSE_LID_T_VIAL.JBI	Close vial
VI_SJ_TRAN_VIAL_T_R2.JBI	Transfer to R2
VI_SJ_PUT_VIAL.JBI	Put vial to table
PE_SJ_PRE_PICK_PIP.JBI	Prepare to pick pipette
PE_SJ_PICK_PIP_0_5.JBI	Pick pipette
PE_SJ_LOAD_1ML_200_100_10_SINGLE.JBI	Load pipette tip
PE_SJ_GET_VI_SINGLE.JBI	Get liquid
PE_SJ_GET_TANK_SINGLE.JBI	Get liquid from tank
PE_SJ_OUT_VI_10ML.JBI	Out liquid to vial 10 mL
PE_SJ_OUT_VI_5ML.JBI	Out liquid 5 mL pipette
PE_SJ_OUT_VI_1ML.JBI	Out liquid 1 mL pipette
PE_SJ_OUT_VI_200.JBI	Out liquid to vial 200 µL pipette
PE_SJ_OUT_VI_100.JBI	Out liquid to vial 100 µL pipette
PE_SJ_OUT_MTP_MULTI.JBI	Multichannel pipette out to MTP
PE_SJ_RELEASE_TIP.JBI	Release tips
PE_SJ_PUT_PIP_0_5.JBI	Put back pipette
TM_SJ_OPEN_LID_F_TM.JBI	Open thermoshaker
TI_SJ_PRE_PICK_LW_F_SHELF.JBI	Prepare to pick labware from shelf
TI_SJ_PICK_LW_F_SHELF.JBI	Pick labware from shelf
TA_SJ_PRE_PUT_F_SH_T_TABLE.JBI	Prepare to put labware to table
TA_SJ_PUT_LW_T_TABLE.JBI	Put labware to table
TM_SJ_PICK_MTP-LID_F_TM.JBI	Transfer labware from thermoshaker to workbench
TM_SJ_PUT_MTP-LID_T_TM.JBI	Transfer labware to thermoshaker
TM_SJ_CLOSE_LID_T_TM.JBI	Close thermoshaker
HO_SJ_PRE_PICK_LW_F_HOTEL.JBI	Prepare to pick labware from hotel
HO_SJ_PICK_LW_F_HOTEL.JBI	Pick labware from hotel
TA_SJ_PUT_LW_T_TABLE.JBI	Transfer labware from hotel
TA_SJ_PUT_RES_LID_T_TABLE.JBI	Unlid reservoir
HO_SJ_PUT_RES_LID_T_RES.JBI	Lid reservoir
TA_SJ_PRE_PICK_LW_F_TABLE.JBI	Prepare to pick labware from table
TA_SJ_PICK_LW_F_TABLE.JBI	Pick labware from table
HO_SJ_PUT_LW_T_HOTEL.JBI	Transfer labware to hotel
LC_SJ_OPEN_DOOR_F_LC.JBI	Open LC
LC_SJ_PUT_MTP_T_DRAWER.JBI	Transfer mtp to tray
LC_SJ_PUT_DRAWER_T_LC.JBI	Transfer tray to LC
LC_SJ_CLOSE_DOOR_T_LC.JBI	Close LC

11.2 Application II: Detection of cholesterol levels in biliary stent samples

Bile is a yellowish fluid secreted in the liver and helps in the breakdown of fats from food. It usually comprises of fats, salts, water, bilirubin and other inorganic salts [124]–[126]. It is stored in the gall bladder until all fats are digested. It flows through the ducts in the liver, gallbladder pancreas to the intestines and carries toxins out of the body. The biliary ducts get blocked due liver damage or acute inflammation and infection due to gallstones or cholesterol stones causing jaundice [127], [128]. Benign tumors and cysts of the gallbladder or liver may also cause obstruction of the duct which prevents the free flow of bile.

The treatment for relieving the blockage depends upon its cause and in cases where the obstruction is irreversible, a metallic or plastic tubular structure [129], [130], known as a stent, is installed surgically to ensure the bile flow. However, the stents have a tendency of clogging (within 5 months to 10 months) increasing the risk of infection build up, recurrent jaundice and chronic liver disease making them ineffective and creating a need for replacement. Analyses of the biliary sludge have revealed bacteria, fungi, fibers, fatty acid crystals, proteins, cholesterol crystals, trace metals and minerals [131] as its primary constituents. Cholesterol was also found inside the stent material leading to the assumption that stent design and material affect the rate of blockage [124]. The role of cholesterol in the encrustation process is not completely understood. Research suggests that the presence of cholesterol may have the ability to limit bacterial growth [132], thus providing a viable solution to frequent polymer stent occlusions.

Hence, a fully automated system using a dual-arm robot was developed to determine the cholesterol levels in solid stent incrustations using gas chromatography mass spectrometry. The measurement method requires a number of time-consuming steps of sample preparation which may lead to fatigue induced manual errors. The robotic process of sample and labware transfer, liquid handling and GC/MS analysis introduces a standard process for sample preparation freeing lab technicians from the repetitive task.

11.2.1 Description of Process

The process of sample preparation for cholesterol analysis is also implemented on RS-2. The starting material and labware are placed in correct positions by the lab technician for handling by the robot. The internal standard (10 ppm concentration) is prepared by dissolving 10 µL 5α-Cholestane (concentration 1,000 ppm) with 3990 µL n-Hexane. This ISTD solution is stored in a 10 mL vial. N-Hexane is also stored in 10 mL vials for later use. A vial of N-Trimethylsilyl-imidazole (TSIM) is placed on the cooling rack on ALP 11 and cooled to 3°C. The TSIM is a derivatization agent. Before the actual sample preparation and analysis is carried out the robot prepares calibration solutions of various known concentrations. Calibration is the measurement of instrument response from a known standard compound. It helps in establishing a reference against which target compounds are compared and evaluated. Calibrations are performed intermittently (typically once a month) to ensure correct measurement results. For this purpose, cholestane (10 ppm) solution is prepared by the lab technician manually by

mixing 40 µL 5α-Cholestane (1,000 ppm) with 3960 µL n-Hexane. A 50 ppm cholesterol solution (1900 µL n-Hexane mixed with 100 µL Cholesterol) is also prepared by the technician. All these materials are prepared and stored in vials manually for the robot to use during the actual sample preparation process. The lab technician places all required labware including pipettes, pipette tips and empty vials in pre-designated positions in the robot working area. The actual sample preparation is carried out by the robot according to the steps given further. An overview of the sample preparation process is shown in Figure 119.

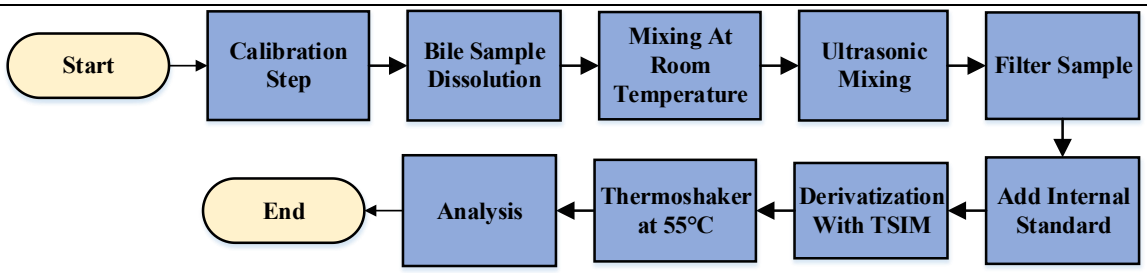


Figure 119: Process flow of cholesterol sample preparation

Calibration: Calibration solutions of 4 different concentrations (10 ppm, 5 ppm, 1 ppm, 0.5 ppm) are prepared. Three samples of each concentration are prepared simultaneously during a single run. The robot mixes the 10 ppm cholestane solution, hexane and 50 ppm cholesterol solution in appropriate proportions. 25 µL TSIM is added to each vial for derivatizing the cholesterol. The samples are derivatized in the thermoshaker for 30 minutes at 55°C before transferring them to the GC-MS for analysis.

The automated sample preparation process is aimed at validating the accuracy of the system. Incrustations built up in a biliary stent are collected, dried and milled to be used for determining the cholesterol content. A solution with a known cholesterol concentration of 8 ppm was used for the determination of the recovery rate. The robotic process is carried according to the following steps:

Dissolution: During the first step 1500 µL cholesterol solution with 8ppm concentration prepared manually is filled in 12 vials of 1.5 mL volume each. For processing real samples, 1 mg of powdered stent incrustation is stored in a vial (vol 1.5 mL); 12 vials are placed on one rack. The vials are filled with 1,000 µL n-Hexane each.

Mixing: A small shaker is used to promote the mixing of the vial contents. The GC vials containing the sample solution are transferred from the ALP to the shaker and shaken for 1 minute at 750 rpm.

Ultrasonic: The mixing and extraction of cholesterol is further promoted by transferring the vials to the ultrasonic bath where they are processed for a further 1 minute.

Filter: When the solution is thoroughly shaken, the vials are transferred back to the ALP. The solution from each vial is filtered into a new vial with the help of a hydrophobic 0.45 µm PVDF filter to remove any remaining solid residue. A volume of 800 µL of the filtered solution is required to create the measurement samples. The liquid transfer is done with the help of a single channel pipette (vol 1 mL). The pipette tips for this pipette cannot be inserted in the 1.5

mL vials for liquid aspiration due to their larger diameter. Hence, the filtered solution is transferred into vials with a volume of 4 mL.

Derivatization: The resultant filtrate (800 µL vol) is pipetted from the 4 mL vials into new GC vials with a volume of 1.5 mL. A volume of 80 µL cholestane solution with a concentration of 10 ppm is added to each of the cholesterol samples as an internal standard. TSIM, which is used as a derivatizing agent is added (20 µL volume) to each of the samples. The TSIM is pipetted with the help of a glass syringe. This solution is then transferred to the thermoshaker to be derivatized at 55°C and 750 rpm for 30 minutes.

Analysis: The vials containing the derivatized solution are transferred to the autosampler of the GC/MS for cholesterol measurement.

Figure 120 represents the actual labware and workbench setup for preparing 12 test samples.

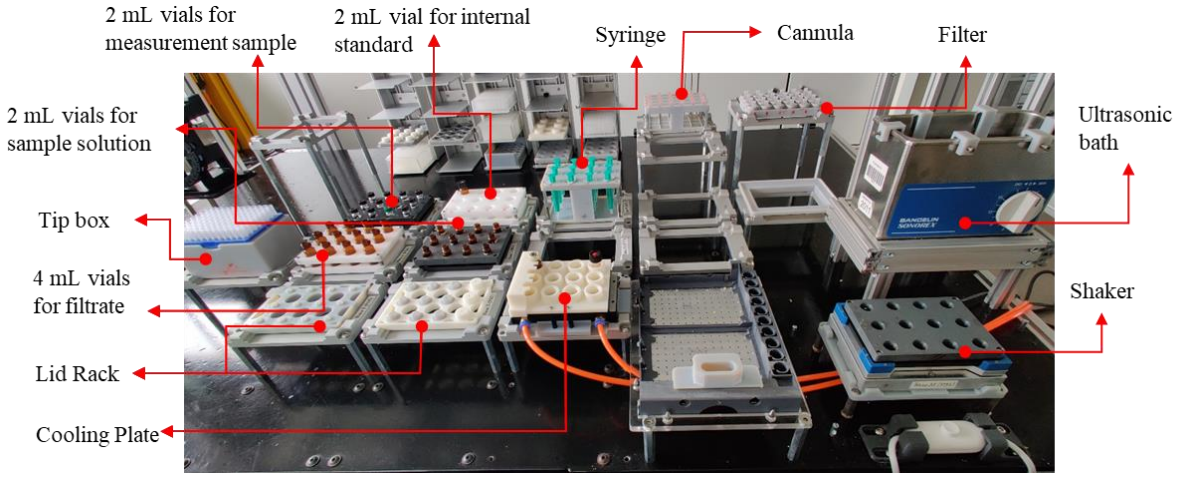


Figure 120: Workbench layout for the cholesterol application

11.2.2 Results and Observations

In order to validate the robotic sample preparation process, 8000 ppb cholesterol solutions were prepared manually and treated as unknown samples. The robot performed all the tasks of filtering and mixing the solutions. The robot added the internal standard and derivatizing agent and transferred the samples to the GC/MS.

Multiple tests were done manually as well as robotically to validate the process and analyze the calibration solutions. The calibration solutions are treated as reference values for finding the unknown concentration of the samples. Table 17 below summarizes the automated calibration results achieved in the RS-2.

Table 17: Table depicting the accuracy of calibration samples as detected by the GC

	Calibration Level *(ISTD 1,000 ppb)*			
	500 ppb	1,000 ppb	5,000 ppb	1,0000 ppb
Cholesterol concentration	1239	1460	4964	9995
CV in %	**5**	**2**	**2**	**4**

The calibration was carried out for 4 concentration levels. Three calibration solutions per concentration level were created and the average values are entered in Table 12. The error was calculated in terms of the coefficient of variation (CV), also known as the relative standard deviation. It is the ratio of the standard deviation to the mean value, expressed as a percentage.

The graph in Figure 121 shows a comparison between the manually and robotically prepared calibration solutions.

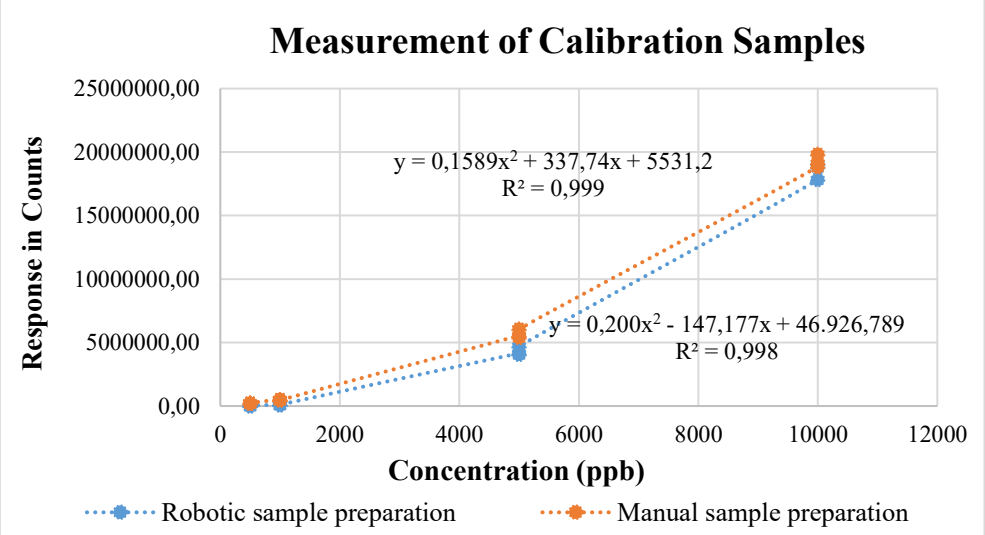


Figure 121: Comparison between manual and robotically prepared calibration solutions for cholesterol analysis

In order to validate the automated process, the sample preparation was carried out eight times and the resultant samples were analyzed in a GC/MS. Each process prepared a set of 12 samples. The initial concentration of cholesterol in the samples was known to be 8,000 ppb. Thus the average detected cholesterol concentration after GC/MS analysis was expected to be 8,000 ppb. However, it was observed that the detected cholesterol concentration ranged from 6856.69 ppb to 9589.40 ppb. The coefficient of variation for each process is below 10%. The results are summarized in Table 18.

Table 18: Sample repeatability

Test	Avg concentration	Standard deviation	Coefficient of variation
1	8056.32 ppb	375.03 ppb	4.66 %
2	9589.40 ppb	389.62 ppb	4.06 %
3	7537.02 ppb	312.51 ppb	4.15 %
4	8066.76 ppb	268.31 ppb	3.33 %
5	7487.27 ppb	138.85 ppb	1.85 %
6	7985.90 ppb	231.01 ppb	2.89 %
7	6856.69 ppb	380.60 ppb	5.55 %
8	9309.13 ppb	449.77 ppb	4.83 %

During all the process runs, large deviation was observed between the response signals of the first and last samples (Table 19). The possibility of the error being introduced during the robotic pipetting stages either due to the air bubbles or residual liquid in the pipette tips resulting in non-identical sample volumes was eliminated by optimizing the pipetting process. It is speculated that the deviation might be attributed to the long idle times of the sample during the preparation process resulting in loss of volatile n-hexane, thus possibly introducing an error

in the concentration levels of the stock solution as well as the internal standard solution. The vial containing the internal standard is opened later in the process. Hence, the evaporation might be less than that of the samples.

Table 19: Table depicting the analysis results pertaining to the concentration of cholesterol in samples as measured by the GC

Sample	Cholesterol Concentration
	ppb
Average	7537
Standard deviation	312
CV in %	**4**

In order to verify the source of the large deviations, a new batch of samples were prepared wherein the internal standard was added to the stock solution prior to filtration. The observations were as depicted in Table 20. It was observed that the coefficient of variation in cholestane and cholesterol is identical, which indicates the changes in concentration of the sample due to evaporation.

Table 20: Measurement values for samples with 5-a-cholestane mixed in stock solution

Sample	Cholesterol Response	5α-Cholestane Response
	counts	***counts***
Average	21044606	6828928
Standard deviation	3757246	1228092
CV in %	**17**	**17**

The time required for various tasks performed by the robot during the sample preparation process were recorded in Table 21. Trained laboratory technicians require approximately 45 minutes to perform one batch of samples. The robot has longer processing times due to intermediate steps such as labware transportation or opening and closing of vials. The automated process uses a range of pre-adjusted variable adjustable pipettes and may require combining multiple pipettes to transfer the required volume which affects the necessary time. However, preparing samples with a high rate of accuracy for a prolonged period of time is highly physically taxing. The technicians are often required to take breaks. Automation allows for increased productivity as the robot can be integrated with other laboratory instruments and can work throughout the day generating and analyzing more samples. It is observed that the robot requires exactly the same amount of time to carry out the sample preparation process each time.

Table 21: Time required by the robot in RS-2 to produce 12 samples

Tasks performed by robot	Time required
	(hh:mm:ss)
Shake vials	00:11:22
Open vials	00:12:41
Filter samples	00:15:36
Pipette sample to new vials	00:02:40
Pipette internal standard to new vials	00:03:00

Addition of derivatizing agent	00:19:26
Close vials	00:10:25
Transfer to thermoshaker	00:01:12
Incubation in thermoshaker	00:30:00
Remove samples from thermoshaker	00: 01:13
Transfer to GC autosampler	00:16:10
Labware transfer	00:02:50
Total time required for the process	**02: 01:26**

11.2.3 Batch Processing of samples

The sample preparation process was designed and implemented for preparing 4 batches generating 48 samples continuously during one run. The syringes and tip boxes were placed in the overhead shelves. Plate type labware such as vial racks, cannulas, filters were placed in the hotels. These labware were treated as resource pools to be used only once per family. The vial lid racks, ultrasonic bath, thermoshaker, shaker and the GC/MS instrument were treated as source labware to be shared by all families. The positions of each labware during the run and before/after the run are detailed in the Table 22.

Table 22: List of all labware required for running four batches of sample preparation with position of each labware before/after and during the run

Labware	Quantity	Storage Position	Position on workbench during run
2 mL vial rack containing 12 vials each	4	Hotels	ALP 1
4 mL vial rack containing 12 vials each	4	Hotels	ALP 4
2 mL vial rack containing 12 samples of stock solution each	4	Hotels	ALP 3
2 mL lid rack	1	Workbench	ALP 5
Syringe rack	4	Shelf	ALP 9
Cannula rack	4	Hotels	GP Base
Filter rack	4	Hotels	ALP 5 mL Tip
1 mL tip box	4	Shelf	ALP 7
300 µL tip box	4	Shelf	ALP 7
TSIM vial	1	Workbench	ALP 11
Glass Syringe	1	Workbench	Temp pipette stand
Thermoshaker	1	Workbench	Thermoshaker
Ultrasonic bath	1	Workbench	US bath
Shaker	1	Workbench	Shaker
GC Autosampler	1	Workbench	GC Autosampler

All the hotel shelves are utilized during the creation of four batches, limiting the number of samples prepared. The labware for each family was brought down to the table before sample preparation. The major obstacles encountered in the batch processing of these samples was the transfer of finished samples to the GC/MS instrument. The autosampler of the GC/MS instrument, due to its design and positioning in the robot environment can only be partially reached by the robot. The robot can only place 30 samples in the GC/MS autosampler. The rest of wells cannot be reached by the robot arm as they lie out of the reachability zone of the robot

arm. The fully extended posture of the robot arm at the GC/MS autosampler wells limits the motion of the arm making it difficult to program the robot to remove old samples and replace them with new samples. In view of these limitations, during batch processing the robot does not load finished samples to the autosampler, instead they are stored in the hotels to be picked up by a technician.

A list of all the motion elements required for this application is given in the Table 23.

Table 23: Table depicting the motions used for sample preparation to detect

Motion Element	Motion performed
VI_SJ_PICK_VIAL.JBI	Pick vial from rack
VI_SJ_PUT_VI_T_SHAKER.JBI	Place vial to shaker
VI_SJ_TURN_ON-OFF_SHAKER.JBI	Turn on shaker
VI_SJ_TURN_ON-OFF_SHAKER.JBI	Turn off shaker
VI_SJ_PICK_VIAL_F_SHAKER.JBI	Pick vial from shaker
VI_SJ_PUT_VIAL_T_US.JBI	Transfer vial to Ultrasonic bath
VI_SJ_US_SET_TIME.JBI	Turn on ultrasonic bath
VI_SJ_PICK_VIAL_F_US.JBI	Pick vial from ultrasonic bath
VI_SJ_PUT_VIAL.JBI	Transfer vials to workbench
VI_SJ_TRAN_VIAL_T_R1.JBI	Transfer to r1
VI_SJ_OPEN_LID_T_VIAL.JBI	Open vials
VI_SJ_PUT_LID_T_RACK.JBI	Put lid to rack
VI_SJ_PICK_LID_F_RACK.JBI	Pick vial lid from rack
VI_SJ_CLOSE_LID_T_VIAL.JBI	Close vial lid
VI_SJ_TRAN_VIAL_T_R2.JBI	Transfer to R2
TI_SJ_PRE_PICK_LW_F_SHELF.JBI	Prepare to pick labware from shelf
TI_SJ_PICK_LW_F_SHELF.JBI	Pick labware from shelf
TA_SJ_PRE_PUT_F_SH_T_TABLE.JBI	Prepare to put labware to table
TA_SJ_PUT_LW_T_TABLE.JBI	Put labware to table
HO_SJ_PRE_PICK_LW_F_HOTEL.JBI	Prepare to pick labware from hotel
HO_SJ_PICK_LW_F_HOTEL.JBI	Pick labware from hotel
HO_SJ_PUT_CANNULA_F_TABLE.JBI	Transfer cannula to workbench
HO_SJ_PUT_FILTER_F_TABLE.JBI	Transfer filter to workbench
SY_SJ_PICK_SYR_F_RACK.JBI	Pick syringe
SY_SJ_PICK_CANULA_F_RACK.JBI	Load cannula
SY_SJ_GET_LIQUID_CANULA.JBI	Aspirate liquid with syringe
SY_SJ_WASTE_CANULA.JBI	Reject cannula
SY_SJ_PICK_FILTER_F_RACK.JBI	Pick filter
SY_SJ_OUT_LIQUID_FILTER.JBI	Dispense liquid
SY_SJ_WASTE_SYRINGE_FILTER.JBI	Reject syringe and filter
PE_SJ_PRE_PICK_PIP.JBI	Prepare to pick pipette
PE_SJ_PICK_PIP_0_5.JBI	Pick pipette
PE_SJ_LOAD_1ML_200_100_10_SINGLE.JBI	Load pipette tip
PE_SJ_GET_VI_SINGLE.JBI	Get liquid
PE_SJ_OUT_VI_1ML.JBI	Out liquid 1 mL
PE_SJ_OUT_VI_100.JBI	Out liquid 100 µL
PE_SJ_RELEASE_TIP.JBI	Reject tip
PE_SJ_PUT_PIP_0_5.JBI	Put back pipette
GP_SJ_PRE_PICK_GP_F_RACK.JBI	Prepare to pick pipette from temporary stand
PE_SJ_PICK_PE_F_TEMP.JBI	Pick glass syringe
PE_SJ_PRE_GET_GS.JBI	Prepare to aspirate
PE_SJ_GET_GS.JBI	Aspirate liquid with glass syringe
PE_SJ_OUT_GS.JBI	Dispense liquid from glass syringe
PE_SJ_WASH_GS.JBI	Wash glass syringe

PE_SJ_PUT_PE_T_TEMP.JBI	Put back glass syringe
TM_SJ_OPEN_LID_F_TM.JBI	Open thermoshaker
TM_SJ_PUT_MTP-LID_T_TM.JBI	Transfer vials to thermoshaker
TM_SJ_CLOSE_LID_T_TM.JBI	Close thermoshaker
TM_SJ_PICK_MTP-LID_F_TM.JBI	Transfer vials to workbench
VI_SJ_PUT_VIAL_T_GC.JBI	Transfer vials to GC

All the motions have been tested more than 10 times and the resultant errors were solved. The results of the various tests and issues solved are detailed in the appendix.

12 Conclusion and Future Scope

In this dissertation, the automated robotic system for sample preparation has been optimized to generate samples for analytical measurement faster. The following goals have been achieved through this dissertation:

1. The performance of an automated system depends on the design of the control software. The concept of motion elements is thus a crucial part. An intelligent re-fragmenting, re-planning and shortening of the existent motion elements was realized in order to make the tasks faster. The node positions (start and end positions of the motions) were changed so that multiple motions share now a node point. The motions were programmed again. In some tasks, motion elements that were solely meant for re-posturing the robot arm and did not handle any labware were eliminated completely. Similarly, multiple motion elements were combined into one so that the two arms moved simultaneously thus saving time. As the number of motion elements reduced, the length of the task jobs was also shortened. The task jobs are .JBI files that are created by the R-Interface 2.0 and sent to the FS100 controller. The task job contains calls to the various motion elements in the correct sequence that is required for the task.
2. A motion that needs to be repeatedly performed on various source and destination positions is programmed with the help of motion frames. The motion frames need a reference point that can be mapped. The reference points were redefined in RS-1 to simplify the motion creation and editing. Separate reference points were defined for the right and the left arm. Thus, the motions of each arm can be adjusted and edited individually without affecting the motions of the other arm. The reference points of the ALPs were changed from an abstract point on the ALP surface to the exact position of one of the vials on a vial adaptor. The changes in reference points have allowed for a standardization of the program structure for a motion element making it easier for a new user to understand the motion element programming.
3. The entire database of motion elements is also saved on the controlling PC. However, robot motion can only be carried out if the motion element is present on the controller. Hence, during the creation of a task job, the R-Interface 2.0 checks to see if the required motion elements are present on the controller. If not, the required motion elements are copied to the controller from the database for the duration of the process. The motion elements are automatically deleted at the end of a process. The programmer needs to ensure, that the latest version of motion elements is saved in the database. This needs to be done manually. As the number of motion elements was reduced in RS-2, the complete database can now be saved into the FS100 controller at a time. This feature of transferring motion elements is particularly useful if new tasks and motions are introduced and the size of database exceeds controller capacity in the future.
4. The various tasks performed by the robot were designed as independent non interacting modules in RS-1. In order to connect the tasks and make the transition seamless, four postures of the robot at four different key points were identified as ‘key points’. The

robot posture at each of the key points is drastically different and the robot cannot travel from one key point to the other directly. Instead, the four key points are connected with the intermediate position. The robot can travel from one key point to other only by first coming to the intermediate position. However, the robot can transition between various tasks that start and end at the same key point. A motion planning algorithm was developed that let the robot controller identify the robot position at the end of each task. Based on the position, the motion planning algorithm decided if the next task can be continued directly or the robot needs to go to the intermediate position to reset the arm posture. If the motion planning algorithm decided to continue to the next task, based on the arm position, it also decides which motion element the task should start from.

5. The motion planning algorithm is implemented in synchronization with the motion search algorithm. The motion search algorithm identifies the attributes of a motion and selects it from the database. The search parameters in the motion search algorithm were changed to make the search shorter. A 'directory' was implemented such that, the R-Interface only recalled the motions pertaining to the task being carried out. Thus, the algorithm had to search through a limited number of motion elements instead of the entire database.
6. In order to increase the number of batches prepared per process run, the structure of the SAMI Ex method was changed. Instead of creating individual methods for the various tasks, only one method file was created for an entire process. The labware was defined as families. Each family is treated as a new set of labware by the SAMI Ex. Thus, a common set of motions can be performed multiple times using different groups of labware. The positions of the labware have to be defined in the SAMI Ex method file. Labware from each group is brought to the workbench and the samples are created. Thus, multiple batches of samples can be prepared during one process run.
7. The end effector system of the robot was replaced by a force detecting gripper system. The motion elements had to be adapted to suit the new grippers. The force feedback allows the grippers to detect when an object has been held. The user needs to specify the grip force and the distance between fingers for each labware. The gripping force changes according to the size and weight of the labware.
8. Two new electronic pipettes were also integrated in the robotic system RS-2. Unlike manual pipettes, where the volumes have to be set manually before the start of the process, aspiration and dispensing volumes of an electronic pipette are set automatically. Thus, one pipette can be used for a range of volume transfers. This feature is useful for complicated liquid transfer tasks, as only limited manual pipettes are available. All volumes may not be able to be pipetted. Also, changing vials multiple times during a process is time consuming. New motions were created for the electronic pipette task. The robot arm is fully extended in order to grab the electronic pipettes. Thus, the arm has a limited range of motion making it challenging to program the pick and place positions accurately. The motions for this task are carried out only by the right arm and the motion elements share the same node points as a manual pipette task. If the two-finger gripper system is replaced with five finger gripper system in the future, the motions for the electronic pipette task can also be used for the manual pipette task performed by a single arm.

9. A solid phase extraction module was installed on the workbench. Motions were planned and programmed for the SPE application. Cartridges and rinsing blocks were transferred from the storage areas to the SPE. In addition, the robot was also programmed to pipette liquid directly on a cartridge placed on the SPE carriage. A special vial adaptor was designed to hold 24 vials of 2 mL volume. The adaptor is designed to be placed below the cartridges to collect the target reagent. The motion planning for the SPE task is critical due to the heavy labware and narrow spaces available to move the arm in order to reach the SPE instrument.
10. Liquid transfer with the help of a glass pipette was newly introduced in RS-2. The glass pipettes are used for transferring accurate volume of liquids from septum protected vials as this cannot be done by a standard manual pipette. The glass syringes, being fragile, need an adaptor for the robot to hold. Thus, an adaptor was designed that can be used for glass vials with volumes ranging from 10 µL to 200 µL. The adaptor is held in the right arm. The left arm holds the syringe plunger and pulls it or pushes it to aspirate or dispense liquid. Thus, new motions were created for handling the glass syringe. The coordination of both the arms is important in this task. In addition, it is necessary to perform this task at a slow speed. As the robot force is large, there is a danger of the needle bending if the piercing motion is performed at a high speed.
11. In order to allow the operator to rectify errors and reset the robot arms to a desired position during an ongoing robotic process, an option for manual move was introduced. The manual move is enabled, if the robot is stopped by the operator by pressing the emergency button or opening the robot enclosure. The robot motion stops. The operator can then rectify errors like a displaced vial lid or a broken vial etc. By enabling the manual move, the operator can enable the teach pendant (which is otherwise disabled during remote processing) while retaining the remote connection to the SAMI Ex and move the robot arms and gripper. After rectification of error, the teach pendant can be put back to the remote mode and the SAMI Ex method can be continued.
12. The two applications of determination of enantiomers in a compound and determination of cholesterol in biliary stent incrustations were performed using the robot. The robot was commanded to perform the sample preparation according to a standard process. The prepared samples were compared with manually prepared samples for accuracy. Multiple batches of samples were prepared to test the repeatability of the robot. The time required by the robot to prepare the samples was recorded.

Further scope also lies in the development of an error handling assistant which the current robotic system lacks. The robot is required to manipulate a number of small objects like the vial lids or pipette tips. The system can give an error if the lids are not correctly picked up by the robot, or the vial accidently falls from the fingers. It is also possible at times, that a pipette tip is fixed too tightly and is not removed to trash in the first attempt. All these cases can lead to major system errors in further tasks and also cause delays if not handled in time. At the same time, such errors are infrequent, hence, employing a person to resolve such issues might be impractical. An error solving assistant that can direct the robot to retrace or repeat the motion

might be helpful. The new retraceable motion elements developed in this system can prove particularly important in such scenarios.

The correctness and accuracy of the current robotic system relies on the accuracy of initial positions and states of the labware provided by the technician. In case of a missing or defective labware, the robot is not capable of detecting errors or inconsistencies. Visual feedback with the help of a camera or infrared signals would help the system detect absent labware without the needing a technician to overlook the process. Image detection will also make the system more flexible. The labware will not have to be placed in fixed positions allowing for a variable layout on the workbench. The positions of all labware are defined in the SAMI Ex method manually by the technician. The flow of labware throughout the process has to be checked during scheduling to avoid collisions. Introducing vision-based detection would further simplify the SAMI Ex method creation.

13 Bibliography

[1] G. Lippi and G. Da Rin, "Advantages and limitations of total laboratory automation: a personal overview," *Clinical Chemistry and Laboratory Medicine (CCLM)*, vol. 57, no. 6, pp. 802–811, 2019.

[2] G. Michalos, K. Dimoulas, K. Mparis, P. Karagiannis, and S. Makris, "A novel pneumatic gripper for in-hand manipulation and feeding of lightweight complex parts—a consumer goods case study," *The International Journal of Advanced Manufacturing Technology*, vol. 97, pp. 3735–3750, 2018.

[3] C. Smith *et al.*, "Dual arm manipulation—A survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[4] C. Yeo and W. Ng, "Automation and productivity in the clinical laboratory: experience of a tertiary healthcare facility," *Singapore Medical Journal*, vol. 59, no. 11, pp. 597–601, 2018.

[5] D. B. Kaber, N. Stoll, K. Thurow, R. S. Green, S. H. Kim, and P. Mosaly, "Human-automation interaction strategies and models for life science applications," *Human Factors and Ergonomics In Manufacturing*, vol. 19, no. 6, pp. 601–621, 2009.

[6] H. Fleischer and K. Thurow, Book "*Automation Solutions for Analytical Measurements: Concepts and Applications*," 1st ed. Weinheim: WileyVCH, 2017.

[7] B. Chandrasekaran and J. M. Conrad, "Human-robot collaboration: A survey," in *SoutheastCon 2015*, vol. June, pp. 1–8, 2015.

[8] I. Holland and J. A. Davies, "Automation in the Life Science Research Laboratory," *Frontiers in Bioengineering and Biotechnology*, vol. 8, pp. 1–18, 2020.

[9] A. Klimchik, A. Ambiehl, S. Garnier, B. Furet, and A. Pashkevich, "Efficiency evaluation of robots in machining applications using industrial performance measure," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 12–29, 2017.

[10] R. Shah and A. B. Pandey, "Concept for Automated Sorting Robotic Arm," *Procedia Manufacturing*, vol. 20, pp. 400–405, 2018.

[11] A. Björnsson, M. Jonsson, and K. Johansen, "Automated material handling in composite manufacturing using pick-and-place systems – a review," *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 222–229, 2018.

[12] S. Makris, P. Tsarouchi, D. Surdilovic, and J. Krüger, "Intuitive dual arm robot programming for assembly operations," *CIRP Annals - Manufacturing Technology*, vol. 63, no. 1, pp. 13–16, 2014.

[13] X. V. Wang, Z. Kemény, J. Váncza, and L. Wang, "Human–robot collaborative assembly in cyber-physical production: Classification framework and implementation," *CIRP Annals - Manufacturing Technology*, vol. 66, no. 1, pp. 5–8, 2017.

[14] S. Makris *et al.*, "Dual arm robot in cooperation with humans for flexible assembly," *CIRP Annals - Manufacturing Technology*, vol. 66, no. 1, pp. 13–16, 2017.

[15] L. Sarkozi, E. Simson, and L. Ramanathan, "The effects of total laboratory automation on the management of a clinical chemistry laboratory. Retrospective analysis of 36 years," *Clinica Chimica Acta*, vol. 329, no. 1–2, pp. 89–94, 2003.

[16] G. R. D. Prabhu and P. L. Urban, "The dawn of unmanned analytical laboratories," *TrAC - Trends in Analytical Chemistry*, vol. 88, pp. 41–52, 2017.

[17] G. Schneider, "Automating drug discovery," *Nature Reviews Drug Discovery*, vol. 17, no. 2, pp. 97–113, 2018.

[18] S. Joshi, X. Chu, K. Ramani, K. Thurow, and H. Fleischer, "Application of a dual arm robot for automated sample preparation for cholesterol determination in biliary stent incrustations," in *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, vol. May, pp. 1–6, 2018.

[19] C. L. Chen, T. R. Chen, S. H. Chiu, and P. L. Urban, "Dual robotic arm 'production line' mass spectrometry assay guided by multiple Arduino-type microcontrollers," *Sensors and Actuators, B: Chemical*, vol. 239, pp. 608–616, 2017.

[20] H. Fleischer, S. Joshi, T. Roddelkopf, M. Klos, and K. Thurow, "Automated Analytical Measurement Processes Using a Dual-Arm Robotic System," *SLAS Technology*, vol. 24, no. 3, pp. 354–356, 2019.

[21] R. D. King *et al.*, "The automation of science," *Science*, vol. 324, no. 5923, pp. 85–89, 2009.

[22] R. Bogue, "Robots in laboratory: a review of applications," *Industrial Robot: An Industrial Journal*, vol. 39, no. 2, pp. 113–119, 2012.

[23] R. Schoop, A. W. Colombo, B. Suessmann, and R. Neubert, "Industrial experiences, trends and future requirements on agent-based intelligent automation," in *IEEE 2002 28th Annual Conference of the Industrial Electronics Society*, vol. 4, pp. 2978–2983, 2002.

[24] S. Manesis and G. Nikolakopoulos, Book "*Introduction to Industrial Automation*," 1st ed. New York: CRC Press, 2018.

[25] R. S. Seaberg, R. O. Stallone, and B. E. Statland, "The Role of Total Laboratory Automation in a Consolidated Laboratory Network," *Clinical Chemistry*, vol. 46, no. 5, pp. 751–756, 2000.

[26] P. Li, Book "*New Trends in Industrial Automation*," 1st ed. London: IntechOpen, 2019.

[27] X. Chu, T. Roddelkopf, H. Fleischer, N. Stoll, M. Klos, and K. Thurow, "Flexible robot platform for sample preparation automation with a user-friendly interface," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2033–2038, 2016.

[28] X. Chu, H. Fleischer, N. Stoll, M. Klos, and K. Thurow, “Application of dual-arm robot in biomedical analysis: Sample preparation and transport,” in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, vol. July, pp. 500–504, 2015.

[29] H. Liu, N. Stoll, S. Junginger, and K. Thurow, “Mobile robotic transportation in laboratory automation: Multi-robot control, robot-door integration and robot-human interaction,” in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 1033–1038, 2014.

[30] H. Liu, N. Stoll, S. Junginger, and K. Thurow, “Mobile Robot for Life Science Automation,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 7, pp. 1–14, 2013.

[31] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, “A New Robust Method for Mobile Robot Multifloor Navigation in Distributed Life Science Laboratories,” *Journal of Control Science and Engineering*, vol. 2016, pp. 1–17, 2016.

[32] S. Neubert *et al.*, “Workflow Management System for the Integration of Mobile Robots in Future Labs of Life Sciences,” *Chemie Ingenieur Technik*, vol. 91, no. 3, pp. 294–304, 2019.

[33] J. Boyd, “Robotic Laboratory Automation,” *Science*, vol. 295, no. 5554, pp. 517–518, 2002.

[34] S. E. F. Melanson, N. I. Lindeman, and P. Jarolim, “How Laboratory Automation Can Help Laboratories, Clinicians, and Patients,” *Laboratory Medicine*, vol. 39, no. 3, pp. 137–143, 2008.

[35] R. S. Seaberg, R. O. Stallone, and B. E. Statland, “The role of total laboratory automation in a consolidated laboratory network.,” *Clinical chemistry*, vol. 46, no. 5, pp. 751–756, 2000.

[36] D. de Ridder, “Artificial intelligence in the lab: ask not what your computer can do for you,” *Microbial Biotechnology*, vol. 12, no. 1, pp. 38–40, 2019.

[37] S. Saitoh and T. Yoshimori, “Fully Automated Laboratory Robotic System for Automating Sample Preparation and Analysis to Reduce Cost and Time in Drug Development Process,” *Journal of Laboratory Automation*, vol. 13, no. 5, pp. 265–274, 2008.

[38] H. Fleischer *et al.*, “Analytical Measurements and Efficient Process Generation Using a Dual–Arm Robot Equipped with Electronic Pipettes,” *Energies*, vol. 11, no. 10, pp. 1–21, 2018.

[39] H. Fleischer *et al.*, “Application of a Dual-Arm Robot in Complex Sample Preparation and Measurement Processes,” *Journal of Laboratory Automation*, vol. 21, no. 5, pp. 671–681, 2016.

[40] D. Wei, M. Li, K. W. King, and L. Yang, “Online and automated sample extraction.,”

Bioanalysis, vol. 7, no. 17, pp. 2227–2233, 2015.

[41] X. Chu, H. Fleischer, T. Roddelkopf, N. Stoll, M. Klos, and K. Thurow, "A LC-MS integration approach in life science automation: Hardware integration and software integration," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, vol. October, pp. 979–984, 2015.

[42] "User Manual: Yaskawa SDA20D," *Yaskawa ELectric Corporation*, 2017. https://www.yaskawa.eu.com/Global Assets/Downloads/Brochures_Catalogues/Robotics/MOTOMAN_Robots/SDA_Series /Flyer_Robot_SDA_Series_E_03.2017.pdf (accessed Feb. 25, 2021).

[43] "Motoman Chair Assembly User Demonstration," *Yaskawa ELectric Corporation*, 2017. https://www.motoman.com/en-us/applications/assembly (accessed Feb. 25, 2021).

[44] H. M. Do, C. Park, and J. H. Kyung, "Dual arm robot for packaging and assembling of IT products," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1067–1070, 2012.

[45] T. H. Connor and M. A. McDiarmid, "Preventing Occupational Exposures to Antineoplastic Drugs in Health Care Settings," *CA: A Cancer Journal for Clinicians*, vol. 56, no. 6, pp. 354–365, 2006.

[46] S. K. Jin Yoshida, Genshin Tei, Chie Mochizuki, Yoshie Masu, Shigeki Koda, "Use of a Closed System Device to Reduce Occupational Contamination and Exposure to Antineoplastic Drugs in the Hospital Work Environment," *The Annals of Occupational Hygiene*, vol. 3, no. 2, pp. 153–160, 2009.

[47] G. Nam *et al.*, "Development of Dual-Arm Anticancer Drug Compounding Robot and Preparation System with Adaptability and High-Speed," *Journal of International Society for Simulation Surgery*, vol. 3, no. 2, pp. 64–68, 2016.

[48] F. F. Schmid *et al.*, "Applicability of a Dual-Arm Robotic System for Automated Downstream Analysis of Epidermal Models," *Applied In Vitro Toxicology*, vol. 2, no. 2, pp. 118–125, 2016.

[49] K. MOORE *et al.*, "Implementation of a High Specification Dual-Arm Robotic Platform to Meet Flexible Screening Needs," *Journal of the Association for Laboratory Automation*, vol. 12, no. 2, pp. 115–123, 2007.

[50] B. CHOI, "Development of Flexible BioRobot Platform for Integrated Clinical Test," *Journal of the Association for Laboratory Automation*, vol. 13, no. 2, pp. 90–96, 2008.

[51] N. L. Alpert, "Automated Instruments for Clinical Chemistry: Review and Preview," *Clinical Chemistry*, vol. 15, no. 12, pp. 1198–1209, 1969.

[52] D. A. Armbruster, D. R. Overcash, and J. Reyes, "Clinical Chemistry Laboratory Automation in the 21st Century - Amat Victoria curam (Victory loves careful preparation).," *The Clinical biochemist. Reviews*, vol. 35, no. 3, pp. 143–153, 2014.

[53] S.-H. Chiu and P. L. Urban, "Robotics-assisted mass spectrometry assay platform

enabled by open-source electronics," *Biosensors and Bioelectronics*, vol. 64, pp. 260–268, 2015.

[54] I. Maurtua, A. Ibarguren, J. Kildal, L. Susperregi, and B. Sierra, "Human-robot collaboration in industrial applications: Safety, interaction and trust," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, pp. 1–10, 2017.

[55] E. Cheung and V. J. Lumelsky, "Motion planning for robot arm manipulators with proximity sensing," in *IEEE International Conference on Robotics and Automation*, pp. 740–745 vol.2, 1988.

[56] C. Thomas, B. Matthias, and B. Kuhlenkötter, "Human - Robot Collaboration – New Applications in Industrial Robotics," *International Conference on Competitive Manufacturing*, pp. 293–299, 2016.

[57] C. Galindo, J. Gonzalez, and J. A. Fernández-Madrigal, "Control architecture for human-robot integration: Application to a robotic wheelchair," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 5, pp. 1053–1067, 2006.

[58] S. El Zaatari, M. Marei, W. Li, and Z. Usman, "Cobot programming for collaborative industrial tasks: An overview," *Robotics and Autonomous Systems*, vol. 116, pp. 162–180, 2019.

[59] M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal, "CoBots: Robust symbiotic autonomous mobile service robots," in *IJCAI International Joint Conference on Artificial Intelligence*, pp. 4423–4429, 2015.

[60] M. Veloso *et al.*, "CoBots: Collaborative robots servicing multi-floor buildings," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5446–5447, 2012.

[61] J. E. Colgate, W. Wannasuphoprasit, and M. A. Peshkin, "Cobots: robots for collaboration with human operators," *AASME International Mechanical Engineering Congress and Exposition*, vol. 58, pp. 433–439, 1996.

[62] P. Akella *et al.*, "Cobots for the automobile assembly line," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, pp. 728–733, 1999.

[63] Å. Fast-Berglund, F. Palmkvist, P. Nyqvist, S. Ekered, and M. Åkerman, "Evaluating Cobots for Final Assembly," *Procedia CIRP*, vol. 44, pp. 175–180, 2016.

[64] W. Wannasuphoprasit, P. Akella, M. Peshkin, and J. E. Colgate, "Cobots: a novel material handling technology," in *American Society of Mechanical Engineers (Paper)*, pp. 1–7, 1998.

[65] "User Manual: ABB dual-arm robots," *ABB Automation Company*, 2020. http://search.abb.com/library/Download.aspx?DocumentID=9AKK106354A3256&LanguageCode=en&DocumentPartId=&Action=Launch (accessed Feb. 25, 2021).

[66] M. Ware, G. Shankar, P. Breslin, and T.-Y. Yang, "Application of automated liquid handling in ligand-binding assay-based bioanalytical method development: the practical considerations," *Bioanalysis*, vol. 10, no. 7, pp. 425–430, 2018.

[67] F. Kong, L. Yuan, Y. F. Zheng, and W. Chen, "Automatic Liquid Handling for Life Science," *Journal of Laboratory Automation*, vol. 17, no. 3, pp. 169–185, 2012.

[68] N. He, T. Liu, and B. Liu, "Technologies and Applications in Micro-Volume Liquid Handling," *Journal of Nanoscience and Nanotechnology*, vol. 16, no. 1, pp. 58–66, 2016.

[69] J. Li *et al.*, Book "*A Modular Liquid Sample Handling Robot for High-Throughput Fourier Transform Infrared Spectroscopy*," 36th ed. Cham: Springer, 2016.

[70] H. Fleischer, D. Baumann, X. Chu, M. Klos, and K. Thurow, "Automated Pipetting in Sample Preparation Processes for Analytical Measurements Using a Dual-arm Robotic System," in *IEEE International Conference on Automation Science and Engineering*, pp. 22–27, 2018.

[71] Integra BioScience AG, "User Manual: Integra Assist Plus," *Integra BioScience AG*, 2020. https://www.integra-biosciences.com/germany-austria/en/pipetting-robots/assist-plus#top (accessed Feb. 25, 2021).

[72] Integra BioScience AG, "Assist Plus Robot Operator Demonstration," *Integra BioScience AG*, 2020. https://www.integra-biosciences.com/germany-austria/en/pipetting-robots/assist-plus (accessed Feb. 25, 2021).

[73] "User Manual: Andrew Pipetting Robot," *Andrew Alliance SA*, 2020. www.andrewalliance.com (accessed Feb. 25, 2021).

[74] Y. Fei, D. Fuqiang, and Z. Xifang, "Collision-free motion planning of dual-arm reconfigurable robots," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 4, pp. 351–357, 2004.

[75] X. Ling, Y. Zhao, L. Gong, C. Liu, and T. Wang, "Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision," *Robotics and Autonomous Systems*, vol. 114, pp. 134–143, 2019.

[76] Wei Lin and XiaoQi Chen, "Dual-arm Cartesian robotic system for parallel tasking," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, pp. 458–463, 2001.

[77] T. Nishi and Y. Mori, "Energy Efficient Motion Planning of Dual-Armed Robots with Pickup Point Determination for Transportation Tasks," in *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 1401–1405, 2018.

[78] H. Tang, Y. Li, and X. Xiao, "A novel flexure-based dual-arm robotic system for high-throughput biomanipulations on micro-fluidic chip," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1531–1536, 2013.

[79] "User Manual: KAWASAKI DuAro dual-arm SCARA robot," *Kawasaki Robotics (USA)*, 2020. https://robotics.kawasaki.com/en1/products/robots/dual-arm-scara/ (accessed Feb. 25, 2021).

[80] "User Manual: Yamaha XY-X Dual-arm Cartesian Robots," *Yamaha Motor Co. Ltd*, 2020. https://global.yamaha-motor.com/business/robot/lineup/xyx/pdf/index/xyx-catalog.pdf (accessed Feb. 25, 2021).

[81] Y. Xiong, V. Shapaval, A. Kohler, J. Li, and P. J. From, "A Fully Automated Robot for the Preparation of Fungal Samples for FTIR Spectroscopy Using Deep Learning," *IEEE Access*, vol. 7, pp. 132763–132774, 2019.

[82] V. Cherezov, A. Peddi, L. Muthusubramaniam, Y. F. Zheng, and M. Caffrey, "A robotic system for crystallizing membrane and soluble proteins in lipidic mesophases," *Acta Crystallographica Section D Biological Crystallography*, vol. 60, no. 10, pp. 1795–1807, 2004.

[83] C. Smith *et al.*, "Dual arm manipulation - A survey," *Robotics and Autonomous Systems*, vol. 60, no. 10. pp. 1340–1353, 2012.

[84] F. Caccavale, V. Lippiello, B. Siciliano, and L. Villani, "RePLiCS: an environment for open real-time control of a dual-arm industrial robotic cell based on RTAI-Linux," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2493–2498, 2005.

[85] P. C. Chen, "Heuristic Sampling: A Method for Predicting the Performance of Tree Searching Programs," *SIAM Journal on Computing*, vol. 21, no. 2, pp. 295–315, 1992.

[86] Dong-Hyung Kim, Sung-Jin Lim, Duck-Hyun Lee, Ji Yeong Lee, and Chang-Soo Han, "A RRT-based motion planning of dual-arm robot for (Dis)assembly tasks," in *IEEE ISR 2013*, pp. 1–6, 2013.

[87] Y. Koga and J.-C. Latombe, "Experiments in dual-arm manipulation planning," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 2238–2245, 1992.

[88] M. Gharbi, J. Cortes, and T. Simeon, "A sampling-based path planner for dual-arm manipulation," in *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 383–388, 2008.

[89] J. Wang, S. Liu, B. Zhang, and C. Yu, "Inverse kinematics-based motion planning for dual-arm robot with orientation constraints," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, pp. 1–14, 2019.

[90] B. Li, J. Gong, Y. Jiang, H. Nasry, and G. Xiong, "ARA*+: Improved path planning algorithm based on ARA*," in *Proceedings - 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2012*, vol. 2, pp. 361–365, 2012.

[91] M. Choi, S. Lim, J. Lee, and C. Han, "Cooperative Path planning for Redundant Dual-arm robot Using Low-dimensional Sample-based algorithm," *IFAC Proceedings*

Volumes, vol. 43, no. 18, pp. 701–708, 2010.

[92] P. Ćurković and B. Jerbić, "Dual-Arm Robot Motion Planning Based on Cooperative Coevolution," *Emerging Trends in Technological Innovation*, vol. 314, pp. 169–178, 2010.

[93] A. Atyabi and D. M. W. Powers, "Review of classical and heuristic-based navigation and path planning approaches," *International Journal of Advancements in Computing Technology (IJACT)*, vol. 5, no. 14, pp. 1–14, 2013.

[94] D. Henrich, C. Wurll, and H. Worn, "Online path planning with optimal C-space discretization," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 3, pp. 1479–1484, 1998.

[95] A. Perez, S. Karaman, A. Shkolnik, E. Frazzoli, S. Teller, and M. R. Walter, "Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4307–4313, 2011.

[96] S. Warren and P. Artemiadis, "On the Control of Human-Robot Bi-Manual Manipulation," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 1, pp. 21–32, 2015.

[97] R. C. Luo, B.-H. Shih, and T.-W. Lin, "Real time human motion imitation of anthropomorphic dual arm robot based on Cartesian impedance control," in *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pp. 25–30, 2013.

[98] I. H. Suh *et al.*, "Design and implementation of a dual-arm robot control system with multi-sensor integrating capability," in *Proceedings IECON '91: 1991 International Conference on Industrial Electronics, Control and Instrumentation*, pp. 898–903, 1991.

[99] C. Yang, P. Liang, Z. Li, A. Ajoudani, C.-Y. Su, and A. Bicchi, "Teaching by demonstration on dual-arm robot using variable stiffness transferring," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1202–1208, 2015.

[100] D. Kruse, J. T. Wen, and R. J. Radke, "A Sensor-Based Dual-Arm Tele-Robotic System," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 4–18, 2015.

[101] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, "Design considerations for safe human-robot collaborative workplaces," in *Procedia CIRP*, vol. 37, pp. 248–253, 2015.

[102] B. Naseri Soufiani and M. A. Adli, "An expanded impedance control scheme for sloshfree liquid transfer by a dual-arm cooperative robot," *Journal of Vibration and Control*, vol. Online, pp. 1–14, 2020.

[103] J. De Schutter *et al.*, "Constraint-based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[104] D. Nicolis, A. M. Zanchettin, and P. Rocco, "Constraint-Based and Sensorless Force Control With an Application to a Lightweight Dual-Arm Robot," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 340–347, 2016.

[105] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[106] Q. Wu, M. Li, X. Qi, Y. Hu, B. Li, and J. Zhang, "Coordinated control of a dual-arm robot for surgical instrument sorting tasks," *Robotics and Autonomous Systems*, vol. 112, pp. 1–12, 2019.

[107] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, and C.-Y. Su, "Finite-Time Convergence Adaptive Fuzzy Control for Dual-Arm Robot With Unknown Kinematics and Dynamics," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 3, pp. 574–588, 2019.

[108] M. Wang, M. Luo, T. Li, and M. Ceccarelli, "A Unified Dynamic Control Method for a Redundant Dual Arm Robot," *Journal of Bionic Engineering*, vol. 12, no. 3, pp. 361–371, 2015.

[109] S. Fang, X. Huang, H. Chen, and N. Xi, "Dual-arm robot assembly system for 3C product based on vision guidance," in *2016 IEEE International Conference on Robotics and Biomimetics, ROBIO 2016*, pp. 807–812, 2016.

[110] J. Krüger, G. Schreck, and D. Surdilovic, "Dual arm robot for flexible and cooperative assembly," *CIRP Annals - Manufacturing Technology*, vol. 60, no. 1, pp. 5–8, 2011.

[111] X. Chu, Thesis""Automation strategies for sample preparation in life science applications,"" University of Rostock, 2016.

[112] X. Chu, H. Fleischer, T. Roddelkopf, N. Stoll, M. Klos, and K. Thurow, "Automated Sample Preparation Using a Dual-Arm Robotic Platform," *American Laboratory*, vol. 48, no. 1, pp. 44–45, 2016.

[113] B. Cohen, S. Chitta, and M. Likhachev, "Single- and dual-arm motion planning with heuristic search," *International Journal of Robotics Research*, vol. 33, no. 2, pp. 305–320, 2014.

[114] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Simultaneous grasp and motion planning: Humanoid robot ARMAR-III," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 43–57, 2012.

[115] M. Likhachev, G. Gordon, and S. Thrun, "ARA•: Anytime A• with provable bounds on sub-optimality," in *Advances in Neural Information Processing Systems*, vol. 16, pp. 767–774, 2003.

[116] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[117] S. Wen and X. Zhu, "Speciation Analysis of Mn(II)/Mn(VII) in Tea Samples Using

Flame Atomic Absorption Spectrometry After Room Temperature Ionic Liquid-Based Dispersive Liquid–Liquid Microextraction," *Food Analytical Methods*, vol. 7, no. 2, pp. 291–297, 2014.

[118] S. Joshi, X. Chu, H. Fleischer, T. Roddelkopf, M. Klos, and K. Thurow, "Analysis of Measurement Process Design for a Dual-Arm Robot Using Graphical User Interface," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, 2019.

[119] N. Ananthi, "Role of Chirality in Drugs," *Organic & Medicinal Chemistry International Journal*, vol. 5, no. 3, pp. 1–6, 2018.

[120] N. M. Davies and X. Wei Teng, "Importance of Chirality in Drug Therapy and Pharmacy Practice: Implications for Psychiatry," *Advances in Pharmacy*, vol. 1, no. 3, pp. 242–252, 2003.

[121] D. Gördes and K. Thurow, "High-Throughput Screening Application For the Determination of Enantiomeric Excess Using ESI-MS," *Journal of the Association for Laboratory Automation*, vol. 11, no. 3, pp. 128–133, 2006.

[122] H. Fleischer and K. Thurow, "Fast mass spectrometry-based enantiomeric excess determination of proteinogenic amino acids," *Amino Acids*, vol. 44, no. 3, pp. 1039–1051, 2013.

[123] H. Fleischer and K. Thurow, "Rapid Enantiomeric Excess Determination of D- and L-Proline Using ESI-MS," *American Laboratory*, vol. 43, no. 9, pp. 32–36, 2011.

[124] G. Donelli, E. Guaglianone, R. Di Rosa, F. Fiocca, and A. Basoli, "Plastic biliary stent occlusion: Factors involved and possible preventive approaches," *Clinical Medicine and Research*, vol. 5, no. 1. pp. 53–60, 2007.

[125] J. J. Y. Sung, "Bacterial biofilm and clogging of biliary stents," *Journal of Industrial Microbiology*, vol. 15, no. 3, pp. 152–155, 1995.

[126] W. Ridtitid and R. Rerknimitr, "Management of an occluded biliary metallic stent.," *World journal of gastrointestinal endoscopy*, vol. 4, no. 5, pp. 157–161, 2012.

[127] C. E. Ruhl and J. E. Everhart, "Gallstone disease is associated with increased mortality in the United States," *Gastroenterology*, vol. 140, no. 2, pp. 508–516, 2011.

[128] D. Barnes, "Gallbladder and Biliary Tract Disease," *Cleaveland Clinic*, pp. 1–4, 2010.

[129] K. Huibregtse, "Plastic or expandable biliary endoprostheses?," *Scandinavian journal of gastroenterology. Supplement*, vol. 200, pp. 3–7, 1993.

[130] U. Seitz, M. Goenka, S. Bohnacker, K. Binmoeller, and N. Soehendra, "Endoscopic biliary stenting: Plastic stent or expandable metallic stent?," *Journal of HBP Surgery*, vol. 4, no. 1, pp. 24–30, 1997.

[131] H. Fleischer, K. Ramani, S. Bauer, M. Warkentin, D. Behrend, and K. Thurow, "Determination of Ca and Trace Metals in Clogged Biliary Endoprostheses Using ICP-MS.pdf," *American Laboratory*, vol. 49, no. 6, pp. 6–9, 2017.

[132] Kozarek RA, “Pancreaticobiliary biofilm: is cholesterol the answer?,” *Gut*, vol. 54, no. 3, pp. 326–328, 2005.

14 Appendix

14.1 Repetitions and success/failure rate for each task

14.1.1 Pipetting with Multi-Channel Pipettes

Test 1: 10 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 2: 100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 3:10 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 4: 300 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 5: 100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 6: 100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 7: 300 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 8:300 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 9: 100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 10: 100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S

Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 11: 100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 12: 100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 13:100 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 14: 10 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 15: 10 µL multi-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

14.1.2 Single- Channel Pipettes

Test 1: 100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 2: 200 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 3: 100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 4: 100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 5: 1 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12

Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 6: 1 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 7: 5 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 8: 10 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 9: 10 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 10: 5 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S

Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 11: 10 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 12: 10 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test13: 200 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 14: 100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 15: 200 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12

Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 16: 100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 17: 200 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 18: 100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 19: 100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 20: 1 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S

Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 21:100 µL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 22: 1 mL single-channel pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

14.1.3 Pipetting with electronic pipettes

Test 1: 1,000 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 2: 1,000 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 3: 1,000 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 4: 200 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 5: 1,000 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 6: 200 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 7: 200 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 8: 200 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 9: 1,000 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

Test 10: 1,000 µL pipette

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Pipette	S	S	S	S	S	S	S	S	S	S	S	S
Load pipette	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Release Tip	S	S	S	S	S	S	S	S	S	S	S	S
Putback pipette	S	S	S	S	S	S	S	S	S	S	S	S

14.1.4 Vial Task

Test 1: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	D	D	S	S	S	S	D	S	S	D	D
Close lid	S	S	S	S	D	D	S	S	S	D	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	D	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 2: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	D	S	S	S	S	S	S	S
Pick lid	S	D	D	S	S	S	S	D	S	S	S	S
Close lid	S	S	S	D	S	S	S	S	S	S	D	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	D	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 3: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	D	S	S	S	S	S
Pick lid	S	S	S	D	S	S	S	D	S	S	D	D
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

The task of picking lids showed consistent errors. The lid rack is elevated during the motion, knocking off remaining lids from the rack. The approach angle and finger position were edited to rectify the error.

Test 4: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	D	S	S	S	S	S	S	S	D
Close lid	S	S	S	S	S	S	D	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S

Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 5: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	D	S	S	S	S	S	S	S	D
Close lid	S	S	S	S	S	S	S	S	S	D	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	D	S	S	S	S	S	S	S	S	D
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 6: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	D	S	S	S	S
Pick lid	S	S	S	D	S	S	S	S	S	S	S	D
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	D	D	S	S	D	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 7: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	D	S	S	S	S	S
Pick lid	D	S	S	S	S	S	S	S	S	S	S	D
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	D	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 8: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	D	D	S	S	S	D	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	D	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S

Transfer to R2	S	S	D	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Transfer of vials from one arm to another failed multiple times, as both the grippers were in an open position simultaneously. A time delay between the actuation of both fingers was introduced.

Test 9: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	D	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	D	S	S	S	S	S	D	D	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

A similar phenomenon to the one above was observed in this case. A time delay between the actuation of both fingers was introduced to ensure a coordinated gripper action.

Test 10: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	D	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 11: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	D	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 12: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S

Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	D	S	S	S	S	S	S	S	D	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 13: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	D	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

In spite of previous changes, a repeated error was observed while picking lids. Two types of lid racks are available for use, one with a curvature to the lid positioning stubs and one without curvature. It is advised to use the labware with curvature to eliminate the error.

Test 14: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 15: 2 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 16: 4 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	D	D
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 17: 4 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 18: 22 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 19: 22 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 20: 10 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12

Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 21: 40 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 22: 10 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 23: 4 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 24: 22 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S

Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 25: 40 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	S
Open lid	S	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

Test 26: 4 mL vials

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick Vial	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R1	S	S	S	S	S	S	S	S	S	S	S	S
Pick lid	S	S	S	S	S	S	S	S	S	S	S	S
Close lid	S	S	S	S	S	S	S	S	S	S	S	D
Open lid	D	S	S	S	S	S	S	S	S	S	S	S
Put lid	S	S	S	S	S	S	S	S	S	S	S	S
Transfer to R2	S	S	S	S	S	S	S	S	S	S	S	S
Put Vial	S	S	S	S	S	S	S	S	S	S	S	S

14.1.5 Syringe Liquid Transfer

Test 1:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	D	S	S	S	D	S	S	S	S	D	S	S
Pick filter	S	S	D	S	D	D	D	D	S	D	D	S
Waste cannula	S	D	S	D	D	D	D	D	S	D	D	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	D	D	D	D	S	D	D	S
Out Liquid	S	S	S	S	D	D	D	D	S	D	D	S

Test 2:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	D	D	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	D	D	D	S	D	D	S	S	S
Waste cannula	S	S	S	D	S	D	S	D	D	D	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	D	S	D	D	S	S	S
Out Liquid	S	S	S	S	S	D	S	D	D	S	S	S

The positioning of the arm with the cannula rack was edited to eliminate the error in loading cannulas to the syringe. The syringe itself was dislodged while aspirating liquid. This caused an error in all the consecutive tasks. The gripper adapter was coated with rubber paint and a layer of sand paper to impart resistive forces.

Test 3:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	D	D	D	S	S	D	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 4:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	D	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	D	S	S	S	D	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

The arm position for ejecting cannula was precisely adjusted.

Test 5:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	D	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	D	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 6:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 7:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	D	S	S	S	D	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	D	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 8:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	D	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 9:

Task	1	2	3	4	5	6	7	8	9	10	11	12

Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 10:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 11:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 12:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 13:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	D	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S

Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 14:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	D	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

The task of picking filter showed intermittent errors. It is concluded, that due to designing tolerances, the filter rack needs to be placed in a specific direction as marked on the labware. Placing the labware in an opposite direction causes the plate to be slightly elevated while loading a filter causing other filters to move.

Test 15:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 16:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 17:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 18:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 19:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Test 20:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S	S	S	S	S	S	S	S	S	S	S	S
Pick cannula	S	S	S	S	S	S	S	S	S	S	S	S
Pick filter	S	S	S	S	S	S	S	S	S	S	S	S
Waste cannula	S	S	S	S	S	S	S	S	S	S	S	S
Waste Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S

14.1.6 Glass Syringe Liquid Transfer

Test 1: with hexane

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe												
Put Syringe												S

Test 2: with hexane

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe												
Put Syringe												S

Test 3: with hexane

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe												
Put Syringe												S

Test 4: with TSIM

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	D									
Out Liquid	S	S	D									
Wash Syringe												
Put Syringe												S

The syringe did not aspirate the intended quantity of liquid. The syringe was blocked causing the plunger to get stuck and the breaking of labware adaptor. To eliminate this error, a new motion of washing the syringe with hexane after every use was introduced.

Test 5: with hexane

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 6: with hexane

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 7: with hexane

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 8: with hexane

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 9: with TSIM (not aspirated because of air bubble in syringe)

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 10: with TSIM

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 11: with TSIM

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S

Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 12: TSIM

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 13: with TSIM

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 14: with TSIM

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

Test 15: with TSIM

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick syringe	S											
Get Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Out Liquid	S	S	S	S	S	S	S	S	S	S	S	S
Wash Syringe	S	S	S	S	S	S	S	S	S	S	S	S
Put Syringe												S

It is advised to wash the syringe manually after every sample preparation run to ensure smooth operation.

14.1.7 Shaker/Ultrasonic task

Test 1:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	D	S	S	S	D	D	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 2:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	D	D	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Robot finger positioning and angle of approach was edited to eliminate the error.

Test 3:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 4:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 5:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 6:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 7:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 8:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 9:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	D	D	S	S	S	D	S	S	S	S	D	D
From US	S	S	S	S	S	S	S	S	S	S	S	S

On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

The plate holding vials in place was deformed with prolonged use. The replacement of the plate re-introduced some positioning errors. The positions were re-adjusted.

Test 10:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 11:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 12:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

Test 13:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick vial	S	S	S	S	S	S	S	S	S	S	S	S
Put to Shaker	S	S	S	S	S	S	S	S	S	S	S	S
From Shaker	S	S	S	S	S	S	S	S	S	S	S	S
Put to US	S	S	S	S	S	S	S	S	S	S	S	S
From US	S	S	S	S	S	S	S	S	S	S	S	S
On/Off Shaker	S	S	S	S	S	S	S	S	S	S	S	S
On/Off US	S	S	S	S	S	S	S	S	S	S	S	S

14.1.8 Thermoshaker Task

Test 1:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 2:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 3:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 4:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 5:

Task	1	2	3	4	5	6	7	8	9	10	11	12

Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 6:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 7:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 8:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 9:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S

Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

Test 10:

Task	1	2	3	4	5	6	7	8	9	10	11	12
Pick rack	S	S	S	S	S	S	S	S	S	S	S	S
Put to TM	S	S	S	S	S	S	S	S	S	S	S	S
From TM	S	S	S	S	S	S	S	S	S	S	S	S
Put to table	S	S	S	S	S	S	S	S	S	S	S	S
On/Off TM	S	S	S	S	S	S	S	S	S	S	S	S
Open TM	S	S	S	S	S	S	S	S	S	S	S	S
Close TM	S	S	S	S	S	S	S	S	S	S	S	S

14.2 List of Variables BXXX used

Variable	Name	Tested Values with Meanings
B000 - B009		keep unchanged
B010	ROOM	1-6, except rooms >3 for hotel 3 and >5 in Hotel 4
B011	HOTEL	0-4
B018	BASE_NO	0, 1, 2, 3, 4, 5, 7, 9, 10, 12,13
B019	VIAL_TYPE	0: 22 mL 1: 10 mL 2: 4 mL 3: 2 mL 4: 40 mL 5: GC-Vial
B020	LW_POS_X	labware x direction
B021	LW_POS_Y	labware y direction
B024	US_X	ultrasonic shaker x direction
B025	US_Y	ultrasonic shaker y direction
B026	X-Cannula / Filter	filter/cannula shaker x direction
B027	Y-Cannula / Filter	filter/cannula shaker y direction
B030	G-P_X	glass pipette x direction
B031	G-P_Y	glass pipette y direction
B032	SHELFNUMBER	tip box bases on shelf: 0-7
B033	GS VOLUME	0: 132 µl in 250 µl Syringe 1: 25 µl in 25 µl Syringe
B034	ONTO_SPE	0: Cartridge on top of Rinsing; 1=Cartridge on top of GC-Vial-Rack
B035	RACK_TYPE	For hotels 0: MTP, tip boxes, syringes 1: 4 mL vials, 2 mL vials, SPE vial block, SPE rinsing plate, cartridges 2: reservoir; 3: 4 part reservoir 4: 8 part reservoir 5: filter rack, vial lid rack 6: cannulas 7: MTP lids
B036	ELECTRIC_PIPETTE	0: 1,000 µL, 1: 200 µL
B037	TEMP_PIP	0,1 (same numbering as shelf)

B038	PRE_WET_REPEAT	
B039	PRE_WET	0: no prewet; 1: prewet
B040	Pipette_1-6	0: 10 mL, 1: 5 mL, 2: 1 mL, 3: 200 µL, 4: 100 µL, 5: 10 µL, 6: 300 µL (multi-) 7: 100 µL (multi-channel) 8: 10 µL (multi-channel)
B043	TIP_X	tips in x direction
B044	TIP_Y	tips in y direction
B050	TANK_X	reservoir x direction
B051	MTP_WELL_X	mtp well in x direction
B052	MTP_WELL_Y	mtp well in y direction
B060	TIME_US	time for US shaker: 1 minute or 5 minutes
B062	Place_GC_Sampler	

14.3 List of PXXX Variables used

Page Nr.	Name		Page Nr.	Name
P000	calc		**P045**	Base 5
P001	calc		**P046**	Base 6
P002	STANDBY POS R1		**P047**	Base 7
P003	STANDBY POS R2		**P048**	Base 8
P004	EMPTY REF R1		**P049**	Base 9
P005	EMPTY REF R2		**P050**	Base 10
P006	GETPOS_R1_TEMP		**P051**	Base 11_R2
P007	GETPOS_R2_TEMP		**P052**	Base 12
P008	R1_TEMP		**P053**	Base 13
P009	R2_TEMP		**P054**	PRE_HO_POS
P010	UNSTABLE		**P055**	PRE_HO_LOWER_POS
P011	SHELF0_CALCREF		**P056**	PRE_BA_POS
P012	HOTEL_CALCREF		**P057**	PRE_SH_POS
P013	PIPETTE_CALCREF		**P058**	EP_MIDDLE_POS
P014	SHELF1_CALCREF		**P060**	Thermoshaker
P015	pick GP		**P061**	INTER POS R1
P016	put GP		**P062**	INTER POS R2
P017	cannula		**P063**	PRE_PE_POS_R1
P018	filter		**P064**	PRE_PE_POS_R2
P019	syringe		**P070**	GS_POS_EMPTY_R1
P020	Base 0_R1		**P071**	GS_POS_EMPTY_R2
P021	Base 1_R1		**P105**	R2_SPE_CALCREF
P022	Base 2_R1		**P106**	R1_SPE_CALCREF
P023	Base3_R1		**P041**	Base 1
P024	Base4_R1		**P042**	Base 2
P025	Base5_R1		**P043**	Base 3
P026	Base6_R1		**P044**	Base 4
P027	Base7_R1		**P045**	Base 5
P028	Base8_R1		**P046**	Base 6
P029	Base9_R1		**P047**	Base 7
P030	Base10_R1		**P048**	Base 8
P031	Base11_R1		**P049**	Base 9
P032	Base12_R1		**P050**	Base 10
P033	Base13_R1		**P051**	Base 11_R2
P034	10 mL pip		**P052**	Base 12

P035	5 mL pip		**P053**	Base 13
P036	ultrasonic		**P054**	PRE_HO_POS
P037	shaker		**P055**	PRE_HO_LOWER_POS
P039	US_SET_TIME_REF		**P056**	PRE_BA_POS
P040	Base 0		**P057**	PRE_SH_POS
P041	Base 1		**P058**	EP_MIDDLE_POS
P042	Base 2		**P060**	Thermoshaker
P043	Base 3		**P061**	INTER POS R1
P044	Base 4		**P062**	INTER POS R2

14.4 List of BPXXX Variables

Page Number	Name	Pulse	Description
BP000	TEMP		do not change
BP001	TEMP	-	do not change
BP002	-	-	do not change
BP003	-	-	do not change
BP004	-	-	do not change
BP005	-	-	do not change
BP006	GETPOS	-	do not change
BP007	GETPOS	-	do not change
BP008	-	-	do not change
BP009	-	-	do not change
BP010	-	-	do not change
BP011	MIDDLE_BASE_POS	95875	Middle frontal robot position for TA, TI Jobs
BP012	HO_BASE_POS	78000	For pick and put from all the hotels
BP013	COOL_PLATE_POS	55388	For positioning above the cooling plate-
BP017	Rack_Canula	46911	Cannula transportation-
BP018	Rack_Filter	32741	Filter transportation
BP020	VIAL	111850	For pick vial and put vial jobs
BP021	VIAL2	95875	For shaker/US/ lid jobs
BP022	SY_IN/OUT_LIQUID	99077	For aspirating and dispensing with syringe
BP030	Glass Pipette	67861	Pick up glass pipette-
BP031	GP_LIQUID	78413	Aspirate / dispense liquid with glass pipette
BP037	US_SET_TIME	61440	US set time job
BP039	THERMOSHAKER	156445	Open / close thermoshaker-
BP040	Pipettes	68358	Pick / place pipette-
BP041	UsingPIP	88657	Aspirate/ dispense liquid with pipette-
BP042	Pipettes Pick 6-8	-69528	Pick/place pipette-

14.5 Settings for each Task in SAMI Ex Run Robot Box

Task	Property	Property Value
Pipette	<Volume></Volume> <Pipette></Pipette> <Channel></Channel> <PrePipette></PrePipette> <PostPipette></PostPipette> <PrePosition></PrePosition> <PostPosition><PostPosition/>	10ML,5,1ML,300,200,100,10 pipette type: 10mL, 5mL,1mL,300uL,200uL,100uL,10uL SINGLE, MULTI null or a number, e.g., "1" null or a number, e.g., "1" current pipette position: 0,1,2,3,4,5,6,7,8, temp0, temp1 position to put pipette: 0,1,2,3,4,5,6,7,8, temp0, temp1
Glass Pipette	<PrePipette></PrePipette> <PostPipette></PostPipette> <BackStandby></BackStandby>	null or a number, e.g., "1" null or a number, e.g., "1" True or False
Electric Pipettes	<Volume></Volume> <PipetteType></PipetteType> <COMPort></COMPort> <PrePipette></PrePipette> <PostPipette></PostPipette> <AspSpeed></AspSpeed> <DisSpeed></DisSpeed> <AddAspVol></AddAspVol> <AddDisVol></AddDisVol>	liquid volume required 1,000, 200 COM8, COM7 null or a number, e.g., "1" null or a number, e.g., "1" 6 6 0 0
Syringe	<Group></Group>	1 or 2
Glass Syringe	<Volume></Volume> <Pipette></Pipette> <Channel></Channel> <PrePipette></PrePipette> <PostPipette></PostPipette> <PrePosition></PrePosition> <PostPosition><PostPosition/>	250, 25 GlassSyringe SINGLE null or a number, e.g., "1" null or a number, e.g., "1" current position of pipette: temp0, temp1 position to put pipette: temp0, temp1
Vial	<Command></Command>	Move, Lid, Delid
Ultrasonic	<Command></Command> <Time></Time>	Move, SetTime 1 or 5
Shaker	<Command></Command>	Move, Press

14.6 List of Motion Elements

Name of Motion Element	Task executed
Electronic Pipettes	
EP_LK_INTO_SPE_OUT	Dispense liquid in SPE cartridge on SPE
EP_LK_OUT_SPE	Move out of SPE cartridge on SPE
EP_SJ_INTO_VIAL_GET	Aspirate liquid from vials on workbench
EP_SJ_INTO_VIAL_OUT	Dispense liquid in vials on workbench
EP_SJ_INTO_TANK_OUT	Dispense liquid in reservoir on workbench
EP_SJ_LOAD	Load pipette tip
EP_SJ_OUT_VIAL	Move out of vial on workbench
EP_SJ_PICKUP	Pick pipette from rack
EP_SJ_PUTBACK	Put pipette to rack
EP_SJ_RELOAD	Prepare to reload tip/ put back pipette
EP_SJ_TIPEJECT	Eject tip
Glass Pipettes	
GP_SJ_AFT_PUT_GP_T_RACK.JBI	Return to intermediate position after release of pipette
GP_SJ_GET_L_F_VIAL.JBI	Aspirate liquid from vial
GP_SJ_OUT_L_T_VIAL.JBI	Dispense liquid into vial
GP_SJ_PICK_GP_F_RACK.JBI	Pick pipette from rack
GP_SJ_PRE_PICK_GP_F_RACK.JBI	Prepare to pick pipette from rack
GP_SJ_PRE_PUT_GP_T_RACK.JBI	Prepare to put pipette to rack
GP_SJ_PUT_GP_T_RACK.JBI	Put pipette to rack for reuse
GP_SJ_RELEASE_GP.JBI	Release pipette to trash rack
Hotel	
HO_BACK_STANDBY.JBI	Move robot to standby from intermediate position
HO_INTER.JBI	Move robot to intermediate position from standby
HO_SJ_PUT_RES_LID_T_RES.JBI	Put lid to MTP or reservoir on workbench
HO_SJ_PUT_RES_LID_T_TABLE.JBI	Remove lid from MTP or reservoir and place on workbench
HO_SJ_AFT_PUT_LW_T_HOTEL.JBI	Return to intermediate position from hotel node
HO_SJ_PICK_LW_F_HOTEL.JBI	Pick labware from hotel
HO_SJ_PRE_PICK_LW_F_HOTEL.JBI	Prepare to pick labware from hotel
HO_SJ_PRE_PUT_LW_T_HOTEL.JBI	Prepare to put labware to hotel
HO_SJ_PUT_LW_T_HOTEL.JBI	Put labware to hotel
HO_SJ_PUT_CANNULA_T_HOTEL.JBI	Put cannula rack to hotel
HO_SJ_PUT_FILTER_T_HOTEL.JBI	Put filter rack to hotel
HO_SJ_PUT_CANNULA_T_TABLE.JBI	Transfer cannula rack from hotel to table
HO_SJ_PUT_FILTER_T_TABLE.JBI	Transfer filter rack from hotel to table
LC Module	
LC_SJ_CLOSE_DOOR_T_LC	Close the LC door
LC_SJ_OPEN_DOOR_F_LC	Open the LC door

LC_SJ_PICK_MTP_F_DRAWER	Transfer labware from LC tray to workbench
LC_SJ_PUT_DRAWER_T_LC	Transfer LC tray to LC
LC_SJ_PUT_DRAWER_T_TABLE	Transfer LC tray from LC to workbench
LC_SJ_PUT_MTP_T_DRAWER	Transfer labware from workbench to LC tray
Manual Pipette	
PE_SJ_BACK_STANDBY_0_5.JBI	Return to intermediate position after placing pipette 0-5
PE_SJ_BACK_STANDBY_6_8.JBI	Return to intermediate position after placing pipette 6-8
PE_SJ_GET_MTP_MULTI.JBI	Aspirate liquid from MTP with multi-channel pipette
PE_SJ_GET_TANK_MULTI.JBI	Aspirate liquid from reservoir with multi-channel pipette
PE_SJ_GET_TANK_SINGLE.JBI	Aspirate liquid from reservoir with single-channel pipette
PE_SJ_GET_VI_SINGLE.JBI	Aspirate liquid from vials with single-channel pipette
PE_SJ_LOAD_1ML_200_100_10_SINGLE.JBI	Load tips to pipette 2-5
PE_SJ_LOAD_300_100_10_MULTI_TIPS.JBI	Lad tips to pipette 6-8
PE_SJ_LOAD_TIP_5_SINGLE.JBI	Load tips to 5 mL pipette
PE_SJ_LOAD_TIP_10ML_SINGLE.JBI	Load tips to 10 mL pipette
PE_SJ_OUT_MTP_100.JBI	Dispense liquid in MTP with 100ul pipette
PE_SJ_OUT_MTP_MULTI.JBI	Dispense liquid in MTP with multi-channel pipette
PE_SJ_OUT_SPE.JBI	Dispense liquid in SPE cartridges on SPE
PE_SJ_OUT_TANK_10_5_1,000_200_100.JBI	Dispense liquid in reservoir with single-channel pipettes
PE_SJ_OUT_TANK_MULTI.JBI	Dispense liquid in reservoir with multi-channel pipette
PE_SJ_OUT_VI_1ML.JBI	Dispense liquid in vials with 1 mL pipette
PE_SJ_OUT_VI_5_ML.JBI	Dispense liquid in vials with 5 mL pipette
PE_SJ_OUT_VI_10.JBI	Dispense liquid in vials with 10ul pipette
PE_SJ_OUT_VI_10ML.JBI	Dispense liquid in vials with 10 mL pipette
PE_SJ_OUT_VI_100.JBI	Dispense liquid in vials with 100ul pipette
PE_SJ_OUT_VI_200.JBI	Dispense liquid in vials with 200ul pipette
PE_SJ_PICK_PE_F_TEMP.JBI	Pick pipette from temporary pipette stand
PE_SJ_PICK_PIP_0_5.JBI	Pick pipette 0-5 from rack
PE_SJ_PICK_PIP_6_8.JBI	Pick pipette 6-8 from rack
PE_SJ_PRE_PICK_PIP.JBI	Prepare to pick pipette
PE_SJ_PUT_PE_T_TEMP.JBI	Put pipette to temporary pipette stand
PE_SJ_PUT_PIP_0_5.JBI	Put pipette 0-5 to rack
PE_SJ_PUT_PIP_6_8.JBI	Put pipette 6-8 to rack
PE_SJ_RELEASE_TIP.JBI	Release tips from pipette
Glass Syringe	
PE_SJ_PRE_GET_GS	Prepare to aspirate liquid with glass syringe
PE_SJ_GET_GS	Aspirate liquid with glass syringe
PE_SJ_OUT_GS	Dispense liquid with glass syringe
PE_SJ_AFT_OUT_GS	After dispensing liquid
PE_SJ_WASH_GS	Wash glass syringe
Pipetting SPE vial block	
PE_GET_VI2GC_S100.JBI	Aspirate liquid form SPE vial block with 100ul pipette

PE_GET_VI2GC_S200.JBI	Aspirate liquid form SPE vial block with 200ul pipette
PE_OUT_VI2GC_S100.JBI	Dispense liquid to SPE vial block with 100ul pipette
PE_OUT_VI2GC_S200.JBI	Dispense liquid to SPE vial block with 200ul pipette
SPE Module	
SP_PICK_F_SPE_T_HO.JBI	Transfer labware from SPE to hotel
SP_PICK_F_SPE_T_SH.JBI	Transfer labware from SPE to shelf
SP_PUT_F_HO_T_SPE.JBI	Transfer labware from hotel to SPE
SP_PUT_F_SH_T_SPE.JBI	Transfer labware from shelf to SPE
Syringe	
SY_SJ_GET_LIQUID_CANULA.JBI	Aspirate liquid with syringe and cannula
SY_SJ_OUT_LIQUID_FILTER.JBI	Dispense liquid with syringe and filter
SY_SJ_PICK_CANULA_F_RACK.JBI	Load cannula from rack to syringe
SY_SJ_PICK_FILTER_F_RACK.JBI	Load filter from rack to syringe
SY_SJ_PICK_SYR_F_RACK.JBI	Pick syringe from rack
SY_SJ_PUT_SYR_T_G1.JBI	Transfer syringe to left arm
SY_SJ_WASTE_CANULA.JBI	Release used cannula to waste bin
SY_SJ_WASTE_SYRINGE_FILTER.JBI	Release used filter syringe to bin
Workbench	
TA_SJ_AFT_PUT_LW_T_TABLE.JBI	Return to intermediate position from workbench node
TA_SJ_PICK_LW_F_TABLE.JBI	Pick labware from workbench
TA_SJ_PRE_PICK_LW_F_TABLE.JBI	Prepare to pick labware from workbench
TA_SJ_PRE_PUT_F_SH_T_TABLE.JBI	Prepare to transfer labware from shelf to workbench
TA_SJ_PRE_PUT_LW_T_TABLE.JBI	Prepare to put labware from hotel to workbench
TA_SJ_PUT_LW_T_TABLE.JBI	Put labware to workbench
Shelf	
TI_SJ_AFTER_PUT_LW_T_SHELF	Return to intermediate position from shelf node
TI_SJ_PICK_LW_F_SHELF.JBI	Pick labware from shelf
TI_SJ_PRE_PICK_LW_F_SHELF.JBI	Prepare to pick labware from shelf
TI_SJ_PRE_PUT_LW_F_TA_T_SH	Prepare to transfer labware from workbench to shelf
TI_SJ_PUT_LW_T_SHELF	Put labware to shelf
Thermoshaker	
TM_SJ_CLOSE_LID_T_TM.JBI	Close lid of thermoshaker
TM_SJ_OPEN_LID_F_TM.JBI	Open lid of thermoshaker
TM_SJ_PICK_MTP-LID_F_TM.JBI	Transfer labware from thermoshaker to workbench
TM_SJ_PUT_MTP-LID_T_TM.JBI	Transfer labware from workbench to thermoshaker
Vials	
VI_SJ_CLOSE_LID_T_VIAL	Close vials with lids
VI_SJ_OPEN_LID_T_VIAL.JBI	Open vials with lids
VI_SJ_PICK_LID_F_RACK	Pick vial lids from rack
VI_SJ_PICK_VIAL.JBI	Pick vials from rack

VI_SJ_PUT_LID_T_RACK	Put vial lids to rack
VI_SJ_PUT_VIAL.JBI	Put vials to rack
VI_SJ_TRAN_VIAL_T_R1.JBI	Transfer vial from right arm to left arm
VI_SJ_TRAN_VIAL_T_R2.JBI	Transfer vial from left arm to right arm
Shaker	
VI_SJ_PICK_VIAL_F_SHAKER.JBI	Pick vial from shaker
VI_SJ_PUT_VIAL_T_SHAKER.JBI	Put vial to shaker
VI_SJ_TURN_ON-OFF_SHAKER.JBI	Press shaker on/ off button
Ultrasonic Bath	
VI_SJ_PICK_VIAL_F_US.JBI	Pick vials from ultrasonic bath
VI_SJ_PUT_VIAL_T_US.JBI	Put vials to ultrasonic bath
VI_SJ_US_SET_TIME	Set bath time by turning knob
GC Module	
VI_SJ_PUT_VIAL_T_GC	put vials to GC autosampler
VI_SJ_AFT_PUT_GC	Return from GC to intermediate position
VI_PRE_PUT_GC.JBI	Transfer vial between arms to change grip orientation

14.7 Checklist and Tips for Robot Job Generation

A few tips need to be kept in mind while teaching new motions or executing the existing jobs to ensure safety of the operator and avoid damage to instruments.

1. GC/MS
    - During the tasks involving loading samples to the GC/MS, the machine should not be running another analysis. The machine should be in 'Ready' or 'Standby' state.
    - The GC-MS rotating arm should be horizontal with respect to the observer as seen from the direction of the safety door.
2. LC/MS
    - Mark the LC/MS position. Double check to see if the position is exactly the same each time before the LC/MS operation.
3. Manual Pipettes
    - Make sure all pipettes are placed in the correct order.
    - Place the 5 mL and 10 mL tip boxes on the workbench while teaching new pipette motions.
    - Teach new motions accordingly to avoid future collisions.
4. Electronic Pipettes
    - Place the 5 mL and 10 mL tip boxes on the workbench while teaching new pipette motions.
    - Teach new motions accordingly to avoid future collisions.
5. Glass Syringe
    - Manually wash the glass syringe before use
    - Always use 2 mL vial with rubber top (not GC vial) for holding washing solution.
    - Always place washing solution in position 4 on the cooling rack.
    - The 5 mL and 10 mL pipette tips should not be on the workbench during this task.
6. Syringe
    - If the syringe is not able to aspirate liquid due to sliding, check the adaptor liner and replace if necessary
    - There should be no tip box/ labware on base 7 during syringe task
    - Place cannula and filter rack with markings facing the robot.
7. Ultrasonic bath
    - Fill ultrasonic bath with Millipore water before use
8. Hotels
    - Do not use rooms 8 of hotels 0-4.
    - Do not user rooms > 3 for hotel 5

Declaration

This dissertation ‘Optimization and Application of a Flexible Dual Arm Robot Based Automation System for Sample Preparation and Measurement’ is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. The work of this dissertation has been done by me under the guidance of Prof. Dr.-Ing. habil. Heidi Fleischer and Prof. Dr. -Ing. habil. Kerstin Thurow, at University of Rostock, Germany. The dissertation has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Shalaka Joshi

Rostock, 25 March, 2021

List of Publications

1. Joshi, S.; Chu, X.; Fleischer, H.; Klos, M.; Thurow, K.: Analysis of measurement process design for a dual-arm robot using graphical user interface. Proceedings, I2MTC 2019, Auckland (New Zealand), 2019
2. Joshi, S.; Chu, X.; Ramani, K.; Thurow, K.; Fleischer, H.: Application of a Dual Arm Robot for Automated Sample Preparation for Cholesterol Determination in Biliary Stents Incrustations. Proceedings, Proceedings, IEEE International Instrumentation and Measurement Technology Conference I2MTC 2018, Houston (US), pp. 1171-1176, 2018
3. H. Fleischer *et al.*, "Analytical Measurements and Efficient Process Generation Using a Dual–Arm Robot Equipped with Electronic Pipettes," *Energies*, vol. 11, no. 10, pp. 1–21, 2018.
4. Fleischer, H; Roddelkopf, T.; Kross, L. A.; Stoll, R.; Joshi, S.; Thurow, K.: Dual-arm Robotic Compound-oriented Measurement System: Integration of a positive Pressure Solid Phase Extraction Unit. Proceedings, I2MTC 2021, (accepted)
5. Fleischer, H.; Joshi, S.; Roddelkopf, T.; Klos, M.; Thurow, K: Automated Analytical Measurement Processes Using a Dual-Arm Robotic System. SLAS Technology, 24(3), pp. 354-356, 2019

Theses

1. A new robot posture, intermediate position, was introduced where both the arms are moved to a posture most conducive for carrying out the designated tasks which helps eliminate a number of task specific motion elements. The robot is moved from the standby position to the intermediate position at the beginning of the process.
2. The motion elements were redesigned and the path was shortened in order to make the processes faster. Some motion elements from the original robotic system are completely eliminated or multiple motions are combined where possible.
3. All the motion elements were characterized by dissimilar node points which are the start and end robot postures in a motion element. The motions could only be combined in a single sequence in the original robotic system.
4. Multiple motion elements have common start and end node points in the optimized robotic system. Motion elements can be combined in a greater number of sequences in order to form new tasks with the help of pre-existing motions.
5. All robot motions now can be stored in the robot controller at a time allowing the robot to perform the full range of possible tasks. The motion files present in the controller are checked by the interface program and any missing files are transferred automatically by the R-Interface during motion planning.
6. The independent structure of motion elements and tasks that require a permanent return to pre-defined point is time consuming and limits the automation effort in the original robotic system.
7. In order to add flexibility to the system and save time, the tasks in the new system are planned by a motion planning algorithm which decides whether, at the end of each task, the robot arms go back to the intermediate position or directly transition to the next task.
8. Four node points are identified as key points. All tasks start and end at one of the key points. The robot can transition directly from one task to another if they share a key point.
9. Both the arms shared reference points defined by the posture of the right arm at various positions in the robot environment making motion calculations pertaining to the left arm complicated in the original robotic system. Changes to motion of the right arm affected the left arm as well.

10. The left and right robot arms serve independent functions and hence have independent reference positions that are required for generating motion frames in the optimized robotic system. Independent reference positions simplify arm motion calculations and motion element editing.
11. The motion element search functionality in original robotic system translated the motion file names into an extended format for the purpose of searching through the database which was insignificant.
12. In order to enhance the job search, a directory is implemented in the optimized robotic system to segregate the motion elements according to the respective tasks reducing the latency of job search from the database.
13. The method creation in SAMI Ex software was changed from individual method files for each task in a process to a single method file for the whole process allowing the user to see the entire process and make editing easier.
14. Groups of labware are defined as families in SAMI Ex that allow uninterrupted preparation of multiple batches of samples. Labware and reagents might require to be refilled manually.
15. The previous gripper system is replaced by force sensing grippers. The new grippers allow the system to detect the presence of an object during transportation tasks.
16. Electronic pipettes of two different volumes have been introduced to the robotic system. The liquid aspiration volumes are set electronically allowing multiple pipetting steps to be carried out consecutively.
17. The task of derivatization using a glass pipette is newly introduced. Glass pipettes are required to transfer liquid from septum protected vials. A new adaptor was designed to hold the glass syringe with the robot gripper.
18. The task of SPE was introduced in the system. The robot can transfer cartridges and vials from the base and storage systems to the SPE carriage. Reagents are pipetted directly into cartridges on the SPE with the help of manual pipettes.
19. The application of determination of chiral enantiomers in a solution was implemented. A time reduction of 30% was observed in the optimized robotic system.
20. The application for determination of cholesterol in biliary incrustations was implemented. The application was successful with the coefficient of variation of below 10%.

Abstract

Robots play a significant role in the automation of life science processes. The high number of degrees of freedom allow the robots to perform tasks analogously to human operators. Every added degree of freedom imparts more mobility to the arm. The two robot arms can be operated in different modes. In the synchronous mode, both the arms can be used jointly in handling of an individual object. In addition, direct transfer of objects from one arm to another is possible without having to set the object down. The two robot arms can also be controlled independently to carry out two different tasks simultaneously. Thus, the programmer can design and plan the various tasks such that multiple tasks can be done simultaneously resulting in a shorter process cycle. This dissertation describes the optimization of the implementation of the Yaskawa SDA10F dual-arm robot to carry out routine sample preparation tasks in a life science laboratory. A large number of movements are required to carry out the individual tasks. The initial version of the control software is characterized by a large number of restrictions. However, the movements can only be applied in a certain order without editing and cannot be used for new applications. The motions created were programmed for certain labware and could access limited positions on the workspace, which limited the output capacity. Standard laboratory equipment often varies slightly in terms of size and weight and might require dissimilar handling tactics. Hence, multiple identical motions were created for different types of labware which created an extensive database. The robot started the motions from a starting position and went back to the starting position after each task. Moreover, the system could only be operated to prepare one batch of samples each time.

In order to overcome the previous limitations, the control concept has been changed. The robotic motions were re-planned such that they can be used with any type of labware such as microtiter plates, reservoirs, vials etc. New motions and positions were defined and implemented to achieve optimal use of the work area for storing labware and to enable the robot to access a larger number of positions on the workbench. This way, new applications can be integrated with the system without the need for additional motion programming making the system flexible. The structure of the robot motion programs has been changed such that a single program can be used for various labware with different requirements making the database manageable and reducing the number of total motions by half. The structure of the initial robot interface program (R-Interface) is also changed so that various tasks of an application are carried out consecutively without the need for the robot to always return to its starting position between tasks. The decision whether or not to continue to the next task is made with the help of an algorithm and is based on the robot posture at the end of each task. The various tasks are independent of each other such that change, addition or elimination of a task from the interface program does not affect the overall system. Such a system can be expanded in the future.

In addition to optimizing the control software, changes were also introduced in the implementation of the task planning and scheduling SAMI Ex software. The initial system required the various tasks to be created as independent files which had to be queued in the correct sequence by the programmer to create a process flow. Such individual files made it

cumbersome to edit the process. The changed implementation of the SAMI Ex involves creating a single process flow file holding details to all the tasks. This way, the entire system can be seen by the user and edited readily. Within the labware required for the process, groups of labware are defined as ‘families’. The sample preparation process can be carried out continuously using just one process flow for as many times as the number of families.

The main result of the work is a time saving processing of the processes. The optimized motions and improved robot interface program (R-Interface 2.0) has been implemented to perform two sample preparation processes. The motions and system improvements were validated by preparing samples for determining the chiral enantiomers in a solution and preparing samples to measure the cholesterol content in biliary incrustations. The processing times were seen to be reduced by 30% in comparison to the initial system design. The processes can now run multiple times continuously to process multiple batches of samples without human intervention without interruption. Both the sample preparation applications were tested for creating four continuous batches with twelve samples each. Labware and reagents may have to be refilled manually at times. The number of samples that can be prepared in bulk is constrained by the amount of labware that can be stored in the robot environment at a time. The new system has thus increased the throughput of the laboratory. The system relieves trained laboratory technicians from the monotonous and stressful job of preparing the samples manually. They can thus direct their efforts towards developing new methods processes.

Zusammenfassung

Roboter spielen eine wichtige Rolle bei der Automatisierung von Life-Science-Prozessen. Doppelarmroboter verfügen aufgrund der hohen Zahl an Freiheitsgraden über die Möglichkeit der Ausführung von Aufgaben analog zum menschlichen Operator. Jeder zusätzliche Freiheitsgrad verleiht dem Roboterarm dabei mehr Beweglichkeit. Die beiden Roboterarme können in unterschiedlichen Modi betrieben werden. Im synchronen Modus übernehmen sie die gemeinsame Handhabung eines einzelnen Objekts, wobei auch die direkte Übergabe von Objekten von einem Arm auf den anderen ohne ein zwischenzeitliches Absetzen möglich ist. Darüber hinaus besteht die Möglichkeit, beide Arme unabhängig voneinander anzusteuern, um zwei verschiedene Aufgaben gleichzeitig auszuführen. Auf diese Weise kann der Programmierer die verschiedenen Aufgaben so entwerfen und planen, dass mehrere Aufgaben gleichzeitig ausgeführt werden, was zu einem kürzeren Prozesszyklus führt.

Die vorliegende Arbeit beschreibt die Optimierung der Implementierung eines Yaskawa-Doppelarmroboters SDA-10F zur Durchführung routinemäßiger Probenvorbereitungs-aufgaben in einem Life-Science-Labor. Für die Durchführung der einzelnen Aufgaben durch den Roboter ist eine Vielzahl an Bewegungen erforderlich. Die Basisvariante der Steuersoftware ist durch eine Vielzahl von Restriktionen gekennzeichnet. Die Bewegungen können jedoch nur in einer bestimmten Reihenfolge angewendet ohne größere Bearbeitung nicht für neue Anwendungen verwendet werden. Die erstellten Bewegungen sind für bestimmte Labware programmiert und können nur auf begrenzte Positionen im Arbeitsbereich zugreifen, wodurch die Ausgabekapazität begrenzt wurde. Standard-Laborgeräte unterscheiden sich häufig geringfügig in Größe und Gewicht und erfordern möglicherweise unterschiedliche Handhabungstaktiken. Daher wurden mehrere identische Bewegungen für verschiedene Arten von Labware erstellt, woraus eine umfangreiche Datenbank resultiert. Der Roboter startet die Bewegungen jeweils von einer Ausgangsposition aus und kehrt nach jeder Aufgabe in die Ausgangsposition zurück. Darüber hinaus konnte das System nur zur Vorbereitung einer Charge von Proben jedes Mal betrieben werden.

Um die vorherigen Einschränkungen zu überwinden, wurde das Konzept der Steuerung verändert. Die Roboterbewegungen wurden so neu geplant, dass sie mit jeglicher Art von Labware wie Mikrotiterplatten, Reservoirs, Vials usw. verwendet werden können. Zusätzlich wurden neue Bewegungen und Positionen definiert und realisiert, um eine optimale Nutzung des Arbeitsbereichs für die Aufbewahrung von Labware zu erreichen und die Zugriffsmöglichkeiten des Roboters auf eine höhere Anzahl an Positionen auf der Werkbank zu ermöglichen. Auf diese Weise können neue Anwendungen in das System integriert werden, ohne dass eine zusätzliche Bewegungsprogrammierung erforderlich ist, wodurch das System flexibel wird. Die Struktur der Roboterbewegungsprogramme wurde so geändert, dass ein einziges Programm für verschiedene Labware mit unterschiedlichen Anforderungen verwendet werden kann, wodurch die Datenbank verwaltbar wird. Die Struktur des Schnittstellenprogramms wird ebenfalls so geändert, dass verschiedene Aufgaben einer Anwendung nacheinander ausgeführt werden, ohne dass der Roboter zwischen den Aufgaben immer zu seiner Ausgangsposition zurückkehren muss. Die Entscheidung, mit der nächsten Aufgabe fortzufahren oder nicht, wird mithilfe eines Algorithmus getroffen und basiert auf der

Roboterhaltung am Ende jeder Aufgabe. Die verschiedenen Aufgaben sind unabhängig voneinander, so dass das Ändern, Hinzufügen oder Entfernen einer Aufgabe aus dem Schnittstellenprogramm keine Auswirkungen auf das Gesamtsystem hat. Ein solches System kann in Zukunft einfach erweitert werden.

Neben der Optimierung der Steuerungssoftware wurden auch Änderungen bei der Implementierung der SAMI Ex-Software zur Aufgabenplanung und -planung vorgenommen. Das ursprüngliche System erforderte, dass die verschiedenen Aufgaben als unabhängige Dateien erstellt wurden, die vom Programmierer in der richtigen Reihenfolge in die Warteschlange gestellt werden mussten, um einen Prozessablauf zu erstellen. Das Durchsuchen einzelner Dateien machte das Bearbeiten des Vorgangs umständlich. Die geänderte Implementierung von SAMI Ex umfasst das Erstellen einer einzelnen Prozessablaufdatei mit Details zu allen Aufgaben. Auf diese Weise kann das gesamte System vom Benutzer gesehen und problemlos bearbeitet werden. Innerhalb der für den Prozess erforderlichen Labware werden Gruppen von Labware als "Familien" definiert. Der Probenvorbereitungsprozess kann kontinuierlich mit nur einem Prozessablauf durchgeführt werden, der so oft wie die Anzahl der Familien ist.

Wesentliches Ergebnis der Arbeiten ist eine Zeiteinsparung bei der Abarbeitung der Prozesse. Die optimierten Bewegungen und das verbesserte Roboterschnittstellenprogramm (R-Interface 2.0) wurden implementiert, um zwei Probenvorbereitungsprozesse durchzuführen. Die Bewegungen und Systemverbesserungen wurden validiert, indem Proben zur Bestimmung der chiralen Enantiomere in einer Lösung und Proben zur Messung des Cholesteringehalts in Galleninkrustationen vorbereitet wurden. Die Verarbeitungszeiten wurden im Vergleich zum ursprünglichen Systemdesign um 30% reduziert. Prozesse können nunmehr mehrere Male kontinuierlich ausgeführt werden, um mehrere Chargen von Proben ohne menschliches Eingreifen ohne Unterbrechung zu bearbeiten. Beide Probenvorbereitungsanwendungen wurden getestet, um vier kontinuierliche Chargen mit jeweils zwölf Proben zu erstellen. Labware und Reagenzien müssen zeitweise manuell nachgefüllt werden. Die Anzahl der Proben, die in großen Mengen vorbereitet werden können, wird durch die Menge an Labware begrenzt, die gleichzeitig in der Roboterumgebung gespeichert werden kann. Das System erhöht den Durchsatz des Labors. Das System entlastet geschulte Labortechniker von der monotonen und stressigen Aufgabe, die Proben manuell vorzubereiten. Sie können so ihre Bemühungen auf die Entwicklung neuer Methodenprozesse richten.