DESIGN CONSIDERATIONS IN HIGH-THROUGHPUT AUTOMATION FOR BIOTECHNOLOGY PROTOCOLS

by

Aura Maria Cardona Baquero

A Dissertation Submitted to the Faculty of

The College of Engineering and Computer Science

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Florida Atlantic University

Boca Raton, FL

December 2014

Copyright 2014 by Aura Maria Cardona Baquero

DESIGN CONSIDERATIONS IN HIGH-THROUGHPUT AUTOMATION

FOR BIOTECHNOLOGY PROTOCOLS

by

Aura Maria Cardona Baquero

This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Zvi S. Roth, Department of Computer & Electrical Engineering and Computer Science, and has been approved by the members of her supervisory committee. It was submitted to the faculty of the College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

SUPERVISORY COMMITTEE:

Zvi Roth, Ph.D.
Dissertation Advisor

Andrew Goldenberg, Ph.D.

Jim Han, Ph.D.

Nurgun Erdol, Ph.D.
Chair, Department of Computer & Electrical Engineering and Computer Science

Oren Masory, Ph.D.

Hanqi Zhuang, Ph.D.

Mohammad Ilyas, Ph.D.
Dean, College of Engineering and Computer Science

Ali Zilouchian, Ph.D.

Deborah L. Floyd, Ed.D.
Interim Dean, Graduate College

Date

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Zvi S. Roth, for his guidance during these years. He has been a true mentor by supporting me during my research. I would also like to express my sincere gratitude to Dr. Andrew Goldenberg for suggesting the research topic; to Dr. Jim Han for taking the time to teach me Arena® and for many suggestions that added more industrial engineering insight; and to the rest of the committee members, Dr. Oren Masory, Dr. Hanqi Zhuang, and Dr. Ali Zilouchian for all their advice and constructive comments of this dissertation.

I would like to thank also Dr. Maria Petrie, to the Latin American and Caribbean Consortium of Engineering Institutions, and T-VEC Technologies for its funding during my studies.

My gratitude also goes to my FAU friends, without their friendship, help, and support this would not have been possible. To my caring and loving family for being the best family anybody could have, I will be forever thankful.

Finally and most important to my loving parents, brothers, and husband there are no words to express how infinitely grateful I am. All my hard work and dedication would not have been possible without your love and support throughout this process.

ABSTRACT

Author: Aura Maria Cardona Baquero

Title: Design Considerations in High-Throughput Automation for Biotechnology Protocols

Institution: Florida Atlantic University

Dissertation Advisor: Dr. Zvi S. Roth

Degree: Doctor of Philosophy

Year: 2014

In this dissertation a computer-aided automation design methodology for biotechnology applications is proposed that leads to several design guidelines.

Because of the biological nature of the samples that propagate in the automation line, a very specific set of environmental and maximum allowed shelf time conditions have to be followed to obtain good yield. In addition all biotechnology protocols require precise sequence of steps, the samples are scarce and the reagents are costly, so no waste can be afforded.

The methodology presented in this dissertation comprises of several distinct procedures. First, the biotechnology laboratory operations, which are the building blocks for implementing biotechnology manual labor protocols, are described and classified in terms of Automation Modules that provide for the automation integration architectures of all the basic operations. The Automation Modules become the building blocks for realizing biotechnology automation protocols. This has been demonstrated on a case study of column chromatography spin technology RNA purification from animal tissue culture cells. Second, the candidate automated configurations resulting from implementing these modules are evaluated to ensure that all timing considerations are met. To this end, two strategies to avoid excessive waiting times of samples in queues have been developed and evaluated with help of a MATLAB® simulator: The first strategy is based on queue clearing and it involves the periodic stopping and resuming of the samples that enter the system. The second strategy involves parallelizing of stations for which queues present severe

risk of product degradation. Parallelizing of stations is also presented as a technique for increasing the overall throughput. Third, after the candidate configurations have been validated, then by using an economic model, a selection criterion based on system throughput, floor space constraints, and equipment cost constraints is use to find the better solution among the plausible ones. The computer-aided design environment adopted for this dissertation is Arena® Software.

The study went a step further and explored the feasibility of developing flexible automation for biotechnology via group technology planning techniques. The feasibility of GT for biotechnology and the design tradeoffs were demonstrated via simple RNA and DNA purification examples.

DESIGN CONSIDERATIONS IN HIGH-THROUGHPUT AUTOMATION

FOR BIOTECHNOLOGY PROTOCOLS

LIST OF TABLES xii

LIST OF FIGURES xiv

1. INTRODUCTION 1

1.1. Overview 1

1.2. Problem Description 3

1.3. Contributions 5

1.4. Dissertation Outline 6

2. MOTIVATION 9

2.1. Complete Manual RNA Purification Protocol 11

2.2. Conceptualization of Manual Production Line 13

2.3. Conceptualization of Automated Production Line 16

3. BIOTECHNOLOGY AUTOMATION LITERATURE REVIEW 19

3.1. Introduction 19

3.2. System Level Approach to Flexible Automation for Biotechnology 20

3.2.1. Flexibility Parameters 21

3.2.2. Laboratory Automation Depending on Labware Manipulation 23

3.3. Customized Automated Solutions 25

3.3.1. ACAPELLA 5K – Liquid Handling System 25

3.3.2. Multiparametric Platform for Analytical Screening of Living Cells 30

3.3.3. HTS-TRAC™ 31

3.3.4. Scripps - High-Throughput Screening for Drug Discovery 33

4. DRUG DISCOVERY 35

4.1. Introduction 35

4.2. The Drug Discovery Process 35
4.3. Essential Components of a Drug Discovery System 38
4.4. High-Throughput Screening 40
4.5. Example: HTS for Drug Discovery at Scripps Florida 42
4.6. HTS for Drug Discovery: Specifications for Upstream Biotechnology Automation Processes – Example Based on the HTS System at Scripps 44
4.7. Role of PCR in Upstream Processes 47
5. UPSTREAM BIOTECHNOLOGY PROTOCOLS 50
5.1. Purification of Total RNA 50
5.1.1. Purification of Total RNA from Animal Cells 52
5.1.2. Purification of Total RNA from Plant Cells 54
5.1.3. Purification of Total RNA from Animal Tissue 56
5.1.4. Purification of Total RNA Summary 57
5.2. Purification of genomic DNA 58
5.2.1. Purification of Genomic DNA from Animal Cells 58
5.2.2. Purification of Genomic DNA from Plant Cells 60
5.2.3. Purification of Gnomic DNA from Animal Tissue 61
5.2.4. Purification of Genomic DNA Summary 61
5.3. Overall Summary (Final Remarks) 62
6. LABORATORY UNIT OPERATIONS (LUO) 64
6.1. Introduction 64
6.2. The 12 Laboratory Unit Operations (LUO) 65
6.2.1. LUO 1: Manipulation 66
6.2.2. LUO 2: Liquid Handling 67
6.2.3. LUO 3: Separation and Purification 68
6.2.4. LUO 4: Conditioning 69
6.2.5. LUO 5: Washing/Drying 71
6.2.6. LOU 6: Agitation 71

6.2.7. LUO 7: Homogenization 72
6.2.8. LUO 8: Breaking/Fragmentation 72
6.2.9. LUO 9: Weighing 73
6.2.10. LUO 10: Measurement and Direct Detection 73
6.2.11. LUO 11: Analysis and Data Extraction 74
6.2.12. LUO 12: Documentation 75
6.2.13. Summary 75
6.3. Classification of LUO Integration Architectures 78
6.3.1. Conceptualization of Automation Modules (AMs) 79
6.3.2. Automation Module 1 (AM1) 83
6.3.3. Automation Module 2 (AM2) 84
6.3.4. Automation Module 3 (AM3) 85
6.3.5. Automation Module 4 (AM4) 87
6.3.6. Automation Module 5 (AM5) 88
6.3.7. Automation Module 6 (AM6) 89
6.3.8. Automation Module 7 (AM7) 89
6.3.9. Summary of Automation Modules 90
6.4. Integration of Automation Modules 91
6.4.1. Complete Example: Preparation of RNA from Tissue Culture Cells 92
6.5. Integration of Available Customized Solutions and Multi-LUO Equipment 98
6.5.1. Complete Table-Top Solutions 98
6.5.2. Integration of Multi-LUO Equipment 100
6.5.3. Discussion 105
6.6. Additional Laboratory Operation 106
6.6.1. Manual Operations 106
6.6.2. Automated Operations 107
7. STRATEGIES FOR QUEUE CLEARING AND PARALLELIZATION IN BIOTECHNOLOGY AUTOMATION DESIGN 110

7.1. Introduction 110
7.2. Validation Tool: MATLAB® Simulator 111
7.3. Definitions, Assumptions, and Basic Relationships 112
7.4. Queue Size Computation 123
7.5. Throughput Calculations 131
7.6. System and Stations Shutdown 132
7.7. Queue Clearing Strategy 133
7.8. Critical Arrival Period 138
7.9. Parallelizing Strategy 139
7.10. Parallelizing Strategy Numerical Example 147
8. DESIGN METHODOLOGY FOR AUTOMATION SUB-SYSTEMS CONFIGURATION SELECTION 151
8.1. Introduction 151
8.2. The Economic Basis for Automation Configuration Selection 153
8.3. Example 1: Five-Station-Process 156
8.3.1. Candidate Configurations for Automation Design 157
8.3.2. Configurations Simulations with Arena® 161
8.3.3. Cost Figures for Equipment Used for the Numerical Example 163
8.3.4. Numerical Example 165
8.4. Example 2: Five-station-Process High-Throughput Implementation 170
8.5. Example 3: Complete RNA Purification Protocol 173
8.5.1. Candidate Configurations for RNA Purification 173
8.5.2. Simulation with Arena® and Selection Ratio Configuration Selection 176
8.6. Summary 179
9. GROUP TECHNOLOGY APPLIED TO BIOTECHNOLOGY 180
9.1. Introduction 180
9.1.1. Code Representation and Part Classification 181
9.1.2. Layout and Flow Analysis 182

9.2. Introduction of Group Technology Implementation in Biotechnology 183
9.2.1. Four Digit Code Representation for RNA and DNA Isolation 186
9.2.2. Part Classification 187
9.2.3. Equipment Layout 190
9.2.4. Validation 191
9.3. Concluding Remarks 193
10. CONTRIBUTIONS AND FUTURE DIRECTIONS 194
10.1. Contributions 194
10.2. Future Directions 196
APPENDICES 200
APPENDIX A 201
APPENDIX B 214
APPENDIX C 221
REFERENCES 230

LIST OF TABLES

Table 4.1 Volumes needed for target screening against the Compound Libraries 46

Table 5.1 Protocols used for Total RNA Purification for different starting materials 57

Table 5.2 Protocols used for Genomic DNA Purification for different starting materials 62

Table 6.1 Summary of LUO methods and integration features 75

Table 6.2 Summary of Automation Modules 91

Table 6.3 Sub-steps for steps 3 to 6 for the theoretical protocol when using only equipment from Group 2 (left), and equipment from Group 2 and 3 (right) 104

Table 7.1 Results for Example 7.1 115

Table 7.2 Results for Example 7.3 122

Table 7.3 Results for Example 7.4 124

Table 7.4 Numerical results for the Parallelizing Strategy 148

Table 8.1 Simulation throughput results for 5-Station protocol 165

Table 8.2 Price of equipment used 165

Table 8.3 Layout results 166

Table 8.4 Results 4 configurations 169

Table 8.5 Comparison results 169

Table 8.6 HT Arena® Simulation results 171

Table 8.7 High-throughput equipment cost 171

Table 8.8 High-throughput cost results 172

Table 8.9 Simulation throughput results 177

Table 8.10 Price of equipment used for RNA Purification protocol 177

Table 8.11 Layout results for RNA Purification 177

Table 8.12 Cost results 178

Table 8.13 Comparison results 178

Table 9.1 Case study protocols – Preparation of RNA and DNA from animal tissue, animal cells, and animal plant cells 185

Table 9.2 Code use for Group Technology implementation 187

Table 9.3 Purification of RNA from different sources 187

Table 9.4 RNA and DNA preparation from tissue-cultured cells 189

Table 9.5 Results comparing a dedicated production line against a flexible line producing only RNA from animal cells 191

Table 9.6 Results comparing two dedicated production line, one producing 1011 and the other one 1012, against a flexible line producing 1011 and 1012 192

LIST OF FIGURES

Figure 2.1 Preparation of RNA cells (QIAGEN, 2006b) 10

Figure 2.2 Conceptual design for the Preparation of RNA from tissue cultured cells production line 15

Figure 3.1 Tower-based automated work-cell conceptual design for magnetic protein complex purification protocol (Najmabadi, 2005, 2006ab). 24

Figure 3.2 Functional schematic of fluid sample handling with the ACAPELLA-5K automated system (Meldrum, 2001ab). 27

Figure 3.3 HTS-TRAC™ Tecan® system (Tecan, 2008) 31

Figure 3.4 Scripps – HTS system comprehensive diagram 33

Figure 4.1 A conceptual block diagram of a complete pharmaceutical research and discovery process (Carnero, 2006) 36

Figure 4.2 Labsystem MultiDrop 384 from Beckman Coulter (Beckman Coulter, 2011) 40

Figure 4.3 FlexStation 3: High efficiency dual monochromator-based optical system for all read modes from Cisbio (CISBIO, 2011) 41

Figure 4.4 Immulux™ 96-well microplate form Dynex (DYNEX, 2011) 41

Figure 4.5 The Scripps Research uHTS platform. a) Industrial anthropomorphic robotic arm, b) pin tool, c) liquid handlers, d) incubators, e) multimode plate reader, and f) kinetic imaging plate reader (Hodder, 2007) 43

Figure 5.1 Process flow chart for Purification of RNA using spin technology 51

Figure 5.2 Disruption method for animal cells: lysis buffer and vortex 52

Figure 5.3 Homogenization method for animal cells: homogenization column and spin technology 53

Figure 5.4 Isolation of RNA: spin technology 54

Figure 5.5 Homogenization method for plant cells: homogenization column and centrifugation 55
Figure 5.6 Disruption and homogenization of animal tissue 56
Figure 5.7 Isolation of Genomic DNA: spin technology 59
Figure 5.8 Homogenization method for plant cells 60
Figure 6.1 Conceptual diagrams for the formulation of AM 2 for filtration 81
Figure 6.2 Arena® model for Automation Module 1 84
Figure 6.3 Arena® model for Automation Module 2 85
Figure 6.4 Arena® model for Automation Module 3 86
Figure 6.5 Arena® model for Automation Module 4 87
Figure 6.6 Arena® model for Automation Module 5 88
Figure 6.7 Arena® model for Automation Module 7 90
Figure 6.8 Arena® model for Preparation of RNA 97
Figure 6.9 DNA Purification System – Thermo Scientific™ (Thermo Scientific, 2014) 99
Figure 6.10 Multi-LUO equipment define as black boxes 100
Figure 6.11 Centro LB 960 Microplate Luminometer (Berthold Technologies, 2014) 102
Figure 6.12 Conceptual block diagram for the theoretical example 103
Figure 6.13 Conceptual block diagram for the Centro LB 960 Microplate Luminomenter for the sequence that will be replace in the theoretical protocol 103
Figure 6.14 Conceptual block diagram for the theoretical protocol after replacing the equipment 104
Figure 6.15 Integration of Centro LB 960 Microplate Luminometer 105
Figure 7.1 Conceptual block diagram of Station *p* 113
Figure 7.2 Conceptual block diagram for an automation line 114
Figure 7.3 *Nin(p,t)* for Example 7.1 116
Figure 7.4 *Nin(p,t)* for a five station system 117
Figure 7.5 *Nin(4,t)* and trend line 118
Figure 7.6 Propagation of the first sample for Example 7.1 120

Figure 7.7 Results for *Nin(p,t)* simulated and calculated using (7.8) for Example 7.1 125

Figure 7.8 Results for *Nin(4,t)* simulated and calculated using (7.9) for Example 7.2 and predicted trend line 126

Figure 7.9 Results for *Nout(p,t)* simulated and calculated using (7.8)..... 128

Figure 7.10 Results for *Nout(4,t)* simulated and calculated using (7.13) for Example 7.2 and predicted trend line 129

Figure 8.1 Subset of operations studied in the example..... 157

Figure 8.2 Conceptual diagram for Configuration 1 159

Figure 8.3 Conceptual diagram for Configuration 2 159

Figure 8.4 Conceptual diagram for Configuration 3 160

Figure 8.5 Conceptual diagram for Configuration 4 161

Figure 8.6 Arena® model for Configuration 1, Configuration 2 and Configuration3 162

Figure 8.7 Arena® model for Configuration 4 162

Figure 8.8 Fanuc M-430iA (FANUC, 2014)..... 163

Figure 8.9 Centrifuge 5427R (Eppendorf, 2013) 164

Figure 8.10 Centrifuge tubes and columns (QIAGEN, 2006b) 164

Figure 8.11 Configuration 1 layout..... 167

Figure 8.12 Configuration 2 layout..... 167

Figure 8.13 Configuration 3 layout..... 168

Figure 8.14 Configuration 4 layout..... 168

Figure 8.15 96-well plate (QIAGEN, 1997) 170

Figure 8.16 96-plate rotor for high-throughput centrifuge (QIAGEN, 1997) 171

Figure 8.17 RNA Purification protocol Configuration 1 layout 174

Figure 8.18 RNA Purification protocol Configuration 2 layout 176

Figure 9.1 Example of OPITZ code (Askin, 1993) 181

Figure 9.2 Conceptual design for the preparation of DNA and RNA from animal cells production line..... 190

1. INTRODUCTION

1.1. Overview

Much industrial experience gained over the years in automation design for automotive, consumer electronics and many other applications has formed the basis of Industrial Engineering now viewed as a mature field of engineering (Groover, 1996). The design of any automation system strives to minimize capital and operational costs as well as floor space and maximize throughput, yield and product variety (Groover, 1996). The latter goal is referred to as Flexible Automation. An automation design is a multi-objective large-scale optimization problem. It thus often requires computer-aided solution. The optimization often starts with a construction of an ad-hoc “feasible solution” that meets a given set of cost, space and throughput constraints and proceeds with incremental design modifications aimed at yielding incremental performance improvements (Singh, 1996).

Biotechnology instrumentation has attracted significant attention in the past few years. Following the completion of genome sequencing of many organisms, and in order to decipher the function of genes and proteins, demand for high-throughput experimentation and protocols became large (Drews, 2000) (Najmabadi, 2006a).

Automation plays an increasingly important role in many aspects of Life Sciences and especially in biotechnology. With advances in automation, the human genome and other genomes have been sequenced. Modern Molecular Biology and biotechnology have created new assays that, when automated, provide larger, more accurate and rapid amounts of information. Similarly, the pharmaceutical industry is heavily dependent on automation, especially as it shifts from products that treat diseases, to analytical methods that detect and classify diseases. Automation for the Life Sciences is therefore broad in scope and includes fluid handling and assay processing, high-throughput screening and drug discovery, high-throughput production and

analysis of protein and DNA microarrays, devices for analyzing living cells, lab-on-a-chip analysis tools, and numerous detection methods (Drews, 2000) (Ausubel, 2002).

Much on-going basic research revolves around specific subsets of such high-throughput automation systems (Carnero, 2006). This includes the study and development of automated lab-on-a-chip, automated systems for liquid handling, fermentation reaction and process automation, genomics and proteomics software automation, DNA and protein micro-array preparation automation, pharmaceutical fabrication and drug screening automation, detection technologies that enable automation for biological processes, automated systems for DNA, proteins, and cell manipulation and analysis, automated scanning program microscopy-based systems for bio-applications, liquid chromatography (LC) and mass spectrometry (MS) bioinstrumentation automation, and system integration including interconnects and interfaces among automated modules (Drews, 2000).

Our chief concern in this dissertation is the Industrial Engineering considerations in the design of the full laboratory automation system for such applications, as was done in (Najmabadi, 2006bc), where Nam Suh's Axiomatic Theory (Suh, 2001) was applied to analyze and design high-throughput automation for both upstream (sample preparation) and downstream (sample analysis) protocols. It was argued that due to the intrinsic constraints in both common architectures found in laboratory automation, which are robotic-based and track-based approaches, the best solution may be to develop a system that combines both architecture approaches for a flexible laboratory system.

In a robotic-based approach a robotic arm moves various labware between different laboratory instruments. Robotic-based systems typically consist of a tabletop platform on which various laboratory modules and instruments are mounted all serviced by a single centrally located robotic arm or by multiple robot manipulators. The robots perform the necessary liquid handling and labware transportation operations (Najmabadi, 2005, 2006a)

A track-based system uses conveyors, rather than robots, as the transportation mechanisms between stations. In such systems modules and instruments are positioned next to the transportation tracks and the containers are moved between stations on the conveyor tracks.

Each station may use feeders to retrieve tubes or other containers from its local storage (Najmabadi 2005, 2006a).

In the past few years, biotechnology research has been carried out in all laboratory sizes (i.e. small-to-medium size laboratories as well as big laboratories/pharmaceutical companies). Great part of the advances made has depended on individual knowledge and expertise that has been developed in small-to-medium size research laboratories (Drews, 2000) (Najmabadi, 2006a). This has centered the attention of the research community in the direction of improving such laboratories' productivity by introducing appropriate automation and high-throughput systems. Small-to-medium size laboratories are not usually economically powerful, and have very limited resources, hence the need for novel flexible and economical solutions.

The scope of the dissertation is narrowed down from Automation design in biotechnology in general to Automation Design of upstream (sample preparation) protocols taking part in biotechnology. We would be interested in the explicit details of automation implementation, down to issues such as the number of robots that are needed, timing constraints, economically-optimized selection of best implementation configurations and likewise questions. This study is carried out keeping in mind small-to-medium size laboratory limitations and constraints. Although, the main concern is small-to-medium size laboratories the theory developed may also be applied to bigger size laboratories.

In particular we will be interested in anything that distinguishes biotechnology automation from automation in other applications.

1.2. Problem Description

In any automation design, regardless of application and discipline, features such as desired throughput, size of parts, tolerances, precision, timing and procedures involved, among many others, have to be taken into consideration (Groover, 1996) (Singh, 1996). It is obvious that automated solutions for automotive production line, electronic assembly, integrated circuit production, and high-throughput biotechnology protocols are very different. The reasons for such differences include the environmental conditions at which the processes involved have to take place (temperature, humidity, cleanliness, etc), the sizes and quantities of the parts and samples,

and the reaction (or response) times for some processes which all may dictate (for instance) different robot end-effectors for different applications (Cardona, 2008).

Let us take for example the fabrication of integrated circuits (IC). IC fabrication has to be done in clean rooms, the equipment employed is very specialized, has tight precision requirements, and most of the steps are intrinsic to IC fabrication (Metz, 2005). Some of the key characteristics of biotechnology applications include the very small sample size (in the order of magnitude of micro liters and less), and extra special care needed to manipulate the samples, steps that are intrinsic to biotechnology protocols, and these lead to the use of specialized equipment.

The most common and basic protocols used in biotechnology are: DNA isolation, construction of recombinant plasmid, PCR, transcription of genomic DNA and analysis of the resulting mRNA, transformation and gene expression, and analysis of DNA and RNA (Ausubel, 2002). These protocols may have common steps that can be automated such as the addition of a given reagent to a sample, mixing the reagents involved, or applying centrifuging steps. However there are also challenging steps for automation ("bottlenecks" as we may say) that may include the way specific pieces of equipment, such as centrifuges and PCR machines, are to be loaded and the time taken for some of such processes to complete (Ngatchou, 2006).

By increasing the throughput in upstream protocols, one is able to process more samples in a shorter amount of time and perform more genomic and proteomic experiments. Reagents are expensive and large amounts of viable samples may not be available. Therefore one of the goals of biotechnology automation is to reduce the amount and size of the reactions. Along with sample volume reduction several benefits such as improved repeatability, quality and efficiency become possible as well.

Any automation design can be cast as a multi-objective multi-constraint optimization problem. The optimization objectives are to increase throughput, reduce samples size (which decreases cost), and develop a flexible system that could perform more than one type of automated protocols and be easily reconfigurable. There are typically multiple constraints such as

limited footprint, a limited budget, and the protocol-specific constraints (i.e. timing, reaction conditions and environmental condition constraints.) (Groover, 1996).

1.3. Contributions

The key contributions of this dissertation are:

Integration of LUOs

Generalized Automation Modules were formulated for the twelve Laboratory Unit Operations (LUOs) used in biotechnology to facilitate the design of automated production lines. By carefully studying the twelve LUOs, common challenges for the integration of the different pieces of equipment used were grouped together. Thereafter, Automation Modules were described and validated. To fully described protocols in terms of LUOs, some manual and automated operations, not taken into consideration in the available literature, were explained.

Automation lines timing and structural considerations

Two strategies to avoid excessive shelf-time for samples that wait in queues are described: The first strategy is based on queue clearing and it involves the stopping and resuming of the samples that enter the system. The second strategy involves parallelizing of stations in which queues present severe risk of product degradation. Parallelizing of stations is also presented as a technique for increasing the overall throughput. A MATLAB® simulator was developed to evaluate the queue clearing and parallelization strategies formulated in this dissertation. This simulator was validated using Arena® software, comparing the results obtained from the MATLAB® simulator to the ones obtained with the Arena® software for the same model, for various models.

Automation sine sub-system implementation selection criterion

An Economic Model was described to help solve the multi-objective problem involving throughput, floor space, and cost constraints. This fiscal-based configuration selection strategy was later use to describe a computer-aided automation design for biotechnology applications using Arena® Software.

Introduction to flexible automation

The feasibility of implementing Group Technology for automation of biotechnology protocols is demonstrated, establishing an initial baseline for future expansion of the idea.

1.4. Dissertation Outline

The dissertation is structured in a way where each chapter reinforces ideas posed (or answers questions raised) in previous chapters. Each chapter also lays the ground for the following chapters. Let us briefly outline the topics covered in the chapters of this dissertation:

Chapter 1 centers on introducing and describing the problem description of this dissertation. This chapter highlights important issues related to biotechnology automation, and explains the scope of the dissertation.

Chapter 2 discusses in great detailed a motivating case study, the first approximation to conceptualize an automated production line from a manual protocol. This motivating case study helps establish some of the issues and questions that will be answered in the following chapters.

Chapter 3 is devoted to literature review of biotechnology automation. This chapter presents relevant examples of recently designed academic and industrial biotechnology automation solutions and explains where the requirements, constraints and issues intrinsic to biotechnology automation are present in current systems. It also describes how these concerns have been dealt with in the past, and assesses the quality of the results obtained.

Chapter 4 consists mostly of literature review of drug discovery. The requirements and specifications that are often used for upstream protocol automation often are dictated by the needs of automated drug discovery campaigns. Drug discovery is nowadays one of the major fields of biotechnology downstream protocols. There is a vast interest in the scientific community to discover and develop new therapeutic drugs for all kinds of diseases by means of understanding better the human genome. Drug discovery has had important breakthroughs in the last 20 years, presently reaching capability of millions of tests that can be performed in very few months. However, the sample preparation automation has often been unable to keep up with

such advances turning upstream processes into a potential bottleneck for the drug development process.

Chapter 5 concludes the three chapter sequence of literature review focusing on the relevant upstream protocols that will be used in the later chapters. The protocols were chosen from a large pool of applications keeping in mind three main attributes: the familiarity with the manual protocol to make its extrapolation to the automation environment possible, the effectiveness of the protocols within the sample preparation for drug discovery context, and the helpfulness of the protocols to demonstrate and explain the different automation issues presented in this study.

Chapter 6 deals with the twelve basic operations used in protocols known as the Laboratory Unit Operations (LUO). How each of these LUOs or consistent groups of LUOs is implemented, is crucial for the complete automation of a biotechnology protocol. Common characteristics for the different methods that carry out the different operations are grouped into seven Automation Modules. The idea behind the introduction of the concept of Automation Modules is to facilitate the design of automated production lines for biotechnology. In particular, an integration of LUOs into Automation Modules is a key step in the development of GT solutions in biotechnology.

Chapter 7 addresses the problem of biological samples that may exceed their maximum allowable shelf times as these wait in various queues associated with the automation line stations. In biotechnology protocols, whenever dealing with biological samples, the time that the samples spend outside a controlled environment is crucial and caution must be taken to avoid low yield due to samples degradation. In this chapter, two strategies to avoid excessive shelf-times are presented. One of these techniques can also be utilized to increase throughput. These strategies are validated via an automation line simulator developed in MATLAB®.

Chapter 8 introduces economic considerations related to biotechnology automation design. The selection of specific component layout configurations, as a sequence of automation line stations are realized, has to be done taking into account cost considerations. The chapter introduces a selection criterion that is based on both the sub-system expected construction cost

along with its expected throughput. The economic model described in Chapter 8, employs Arena® software to help solve this multi-objective optimization problem.

Increasing the capability for biotechnology laboratories to be able to perform different protocols and produce different samples is very important for increasing their productivity. One plausible solution to increase the laboratories flexibility, therefore, productivity, is the implementation of Group Technology. **Chapter 9** explains the main concepts of Group Technology, and discusses the feasibility of implementing Group Technology to automate biotechnology protocols.

Chapter 10 describes potential future research directions that arise from this study. In the development of the various dissertation contributions, many simplifying assumptions where made. The relaxing of such assumptions may potentially lead to new directions. The main simplifying assumptions throughout this dissertation were: the operation of each automation line station in open loop using no feedback sensors to monitor each station's yield; the protocols demonstrated in this dissertation are only a small sample from a very large pool of available protocols; and only off-the-shelf equipment is used in the study. The algorithms developed for mitigating some of the automation line timing constraints neglected the transportation time between stations. These results can be generalized. For convenience of control and prediction of performance certain automation line components were assumed to run synchronized. Other potential future directions arise from issues that were deliberately not covered by this study, such as, the introducing of labeling steps to the system to enable tracking and control of samples going through equipment that is shared among several stations, and to enable protocols that may require branching of the lines based on real time testing of the products.

2. MOTIVATION

This chapter presents a typical biotechnology process that can serve as a starting point for this study. The process consists of a group of experiments developed for the Florida Atlantic University BSC 4403 Biotechnology Lab 1 course (Florida Atlantic University, 2006). The overall goal of these experiments is the preparation of RNA from tissue-cultured cells (Figure 2.1). In this study the experiments are first laid out conceptually as a manual-labor production line. This production line representation, with its sequential stages, serves as a basis for further deductive process as to how each station may be automated. The process of conversion to automation involves the identification of equipment needed for the various process steps. Design consideration issues raised in this motivating example will follow us throughout this dissertation.

The protocol for preparation of RNA from tissue-cultured cells described in this section claims to do Purification of Total RNA. Total RNA is understood as all RNA molecules available within a cell regardless of their type. In reality, due to the binding capabilities of the RNeasy column used in the protocol, only molecules longer than 200 nucleotides are purified. RNA molecules such as 5.8S rRNA, rRNA, and tRNA are therefore excluded. These smaller molecules compose of about 15% to 20% of total RNA. This protocol is described in this chapter and later in a more depth in Chapter 5. It is important to clarify that after the Preparation of RNA procedure the available RNA have not yet been cut/divided or replicated into specific desired segments. Post-processing is needed to get the desired segments in sufficient quantity by means of PCR (QIAGEN, 1997, 2001a, 2003, 2006b).

The goal of this case study process is the Isolation of RNA from Drosophila Schneider 2 cells that were transfected with a plasmid expressing dact mRNA (RE37047) (Florida Atlantic University, 2006). By transfecting the cells, the expression of the above specific gene is guaranteed to ensure that the desired mRNA, RE37047 in this case, is present.

This specific set of educational experiments require steps that employ proprietary sample-processing devices and chemicals such as QIAshredder columns, RNeasy columns (Figure 2.1), Lysis buffer RLT, B-mercaptoethnol, 70% ethanol, RW1 buffer, Wash buffer RPE, and RNase free water, all contained in the RNeasy Mini - QIAshredder Kit, available from QIAGEN Inc. (QIAGEN, 2002, 2006b).

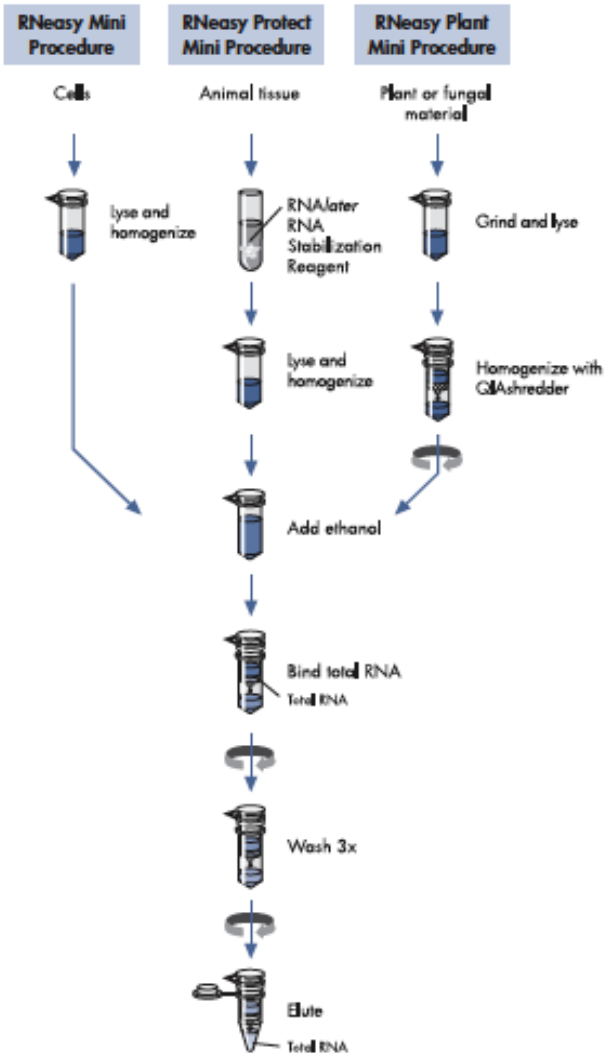

Figure 2.1 Preparation of RNA cells (QIAGEN, 2006b)

The intent of this chapter is to state the motivating issues that arise from this case study. This is why at this point in time only the steps description of the protocol is presented. In a later chapter (Chapter 5), a full detailed description and explanation of the protocol and its elements is presented.

2.1. Complete Manual RNA Purification Protocol

Below is a summary of the steps of the experiment. This experiment is designed to be executed manually by a single operator who retrieves from a storage a 2ml tube filled with frozen cells and then works through all the process steps to obtain at the end the RNA suspended in 50µl of RNase free water. As all tubes, columns and buffers are part of a commercial kit provided by QIAGEN. The lab quantities, timings, and speeds are all specified by the kit manufacturer, and all steps are carried out in 2ml collection tubes. Also, unless otherwise stated the operations are carried out at room temperature, between 18ºC to 26ºC. An expert operator can execute the protocol in approximately 20 minutes.

1. Retrieval of one tube full of frozen cells.
2. Addition of 350 µl of RLT buffer followed by gentle vortexing until cells are lysed.
3. Transferring of the lysate into a QIAshredder spin column placed in a 2ml collection tube.
4. Centrifuging of the tube for 2 minutes at 14,000 rpm in a 4ºC incubator.
5. Addition of 350 µl of 70% ethanol to the lysate in the collection tube.
6. Transferring of the combined 700µl, including any precipitate that may have formed, to an RNeasy mini column in a 2ml collection tube.
7. Centrifuging of the tube for 15 seconds at 14,000 rpm.
8. Discarding of the flow-through and reattaching the collection tube to the mini column.
9. Addition of 700µl of buffer RW1 to the mini-column
10. Centrifuging of the tube for 15 seconds at 14,000 rpm.
11. Discarding of the flow-through and the collection tube.
12. Transferring of the RNeasy mini column to a new 2ml collection tube.
13. Addition of 500µl of the wash buffer RPE into the column.
14. Centrifuging of the tube for 15 seconds at 14,000 rpm.
15. Discarding of the follow-through and reattaching the collection tube to the mini column.
16. Addition of another 500µl of the wash buffer RPE into the column.

17. Centrifuging of the tube for 2 minutes at 14,000 rpm.
18. Addition of 50μl of RNase free water and attaching a new collection tube.
19. Elution of the RNA from the column by centrifugation for 2 minutes at 14,000 rpm.
20. Storage of the collecting tube at -20°C in a freezer.

In order to conceptualize the above experiment protocol as a manual-labor production line, the steps mentioned above have to be broken down into smaller more basic tasks.

1. Picking up a tube filled with frozen cells.
2. Adding a specific amount of RLT buffer
3. Vortexing the tube
4. Transferring the lysate into a QIAshredder spin column
5. Coupling the QIAshredder spin column into a collection tube of a specific size
6. Centrifuging for a given amount of time at a given speed and at a given ambient temperature
7. Adding a specific quantity of 70% ethanol to the lysate in the collection tube.
8. Discharging the QIAshredder spin column
9. Transferring the combined solution, including any precipitate that may have formed, to a RNeasy mini column
10. Coupling the RNeasy spin column into a collection tube of a specific size
11. Centrifuging for a given amount of time at a given speed and at a given ambient temperature
12. Discarding of the flow-through and collection tube
13. Attaching a new collection tube to the RNeasy mini column.
14. Adding a specific quantity of buffer RW1 to the RNeasy mini column
15. Centrifuging for a given amount of time at a given speed and at a given ambient temperature
16. Discarding of the flow-through and the collection tube.
17. Attaching a new collection tube to the RNeasy mini column.
18. Adding a specific quantity of the wash buffer RPE into the RNeasy mini column.

19. Centrifuging for a given amount of time at a given speed and at a given ambient temperature
20. Discarding the follow-through and collection tube
21. Attaching a new collection tube to the RNeasy mini column.
22. Adding a specific quantity of the wash buffer RPE into the column.
23. Centrifuging for a given amount of time at a given speed and at a given ambient temperature
24. Discarding the follow-through and collection tube
25. Adding a specific quantity of RNase free water
26. Attaching a new collection tube to the RNeasy mini column.
27. Eluting the RNA from the column by centrifugation for a given amount of time at a given speed and at a given ambient temperature
28. Storing the collecting tube at -20°C in a freezer.

2.2. Conceptualization of Manual Production Line

The 28 manual labor tasks described above are first set up as a manual labor production line. These tasks can be arranged in 17 sequential stations where a single human operator tends each station (Figure 2.2). It appears that only one storage unit (a freezer) is needed. This storage place holds the tubes with frozen cells and this is where we could store the final tube that contains the extracted RNA. It is assumed that there is a rack located between stations where the tube prepared by one-operator is left to the operator of the next stage.

The resulting production line stations are:

1. Picking up sample from storage (Task 1)
2. Adding RLT Buffer (Task 2)
3. Vortex (Task 3)
4. Transferring into spin mini-column (Tasks 4 and 5)

   Station includes the storage of new spin mini-columns.
5. Centrifuging (Task 6)
6. Adding ethanol (Tasks 7 and 8)

Station includes the storage of spin mini-columns

7. Transferring into mini-column (Tasks 9 and 10)

   Station includes storage of the mini-columns and collection tubes.

8. Centrifuging (Task 11)
9. Discarding flow-through, attaching collecting tube, and adding RW1 (Tasks 12, 13, and 14)

   Station includes storage for the new collection tubes and disposal place for the used collection tubes.

10. Centrifuging (Task 15)
11. Discarding flow-through, attaching collection tube, and adding RPE (Tasks 16, 17, and 18)

    Station includes storage for the new collection tubes and disposal place for the used collection tubes.

12. Centrifuging (Task 19)
13. Discarding flow-through, attaching collection tube, and adding RPE (Tasks 20, 21, and 22)

    Station includes storage for the new collection tubes and disposal place for the used collection tubes.

14. Centrifuging (Task 23)
15. Discarding flow-through, attaching collection tube, and adding 50µl of RNase free water (Tasks 24, 25, and 26)

    Station includes storage for the new collection tubes and disposal place for the used collection tubes.

16. Centrifuging (Task 27)
17. Storing of the collecting tube (Task 28)

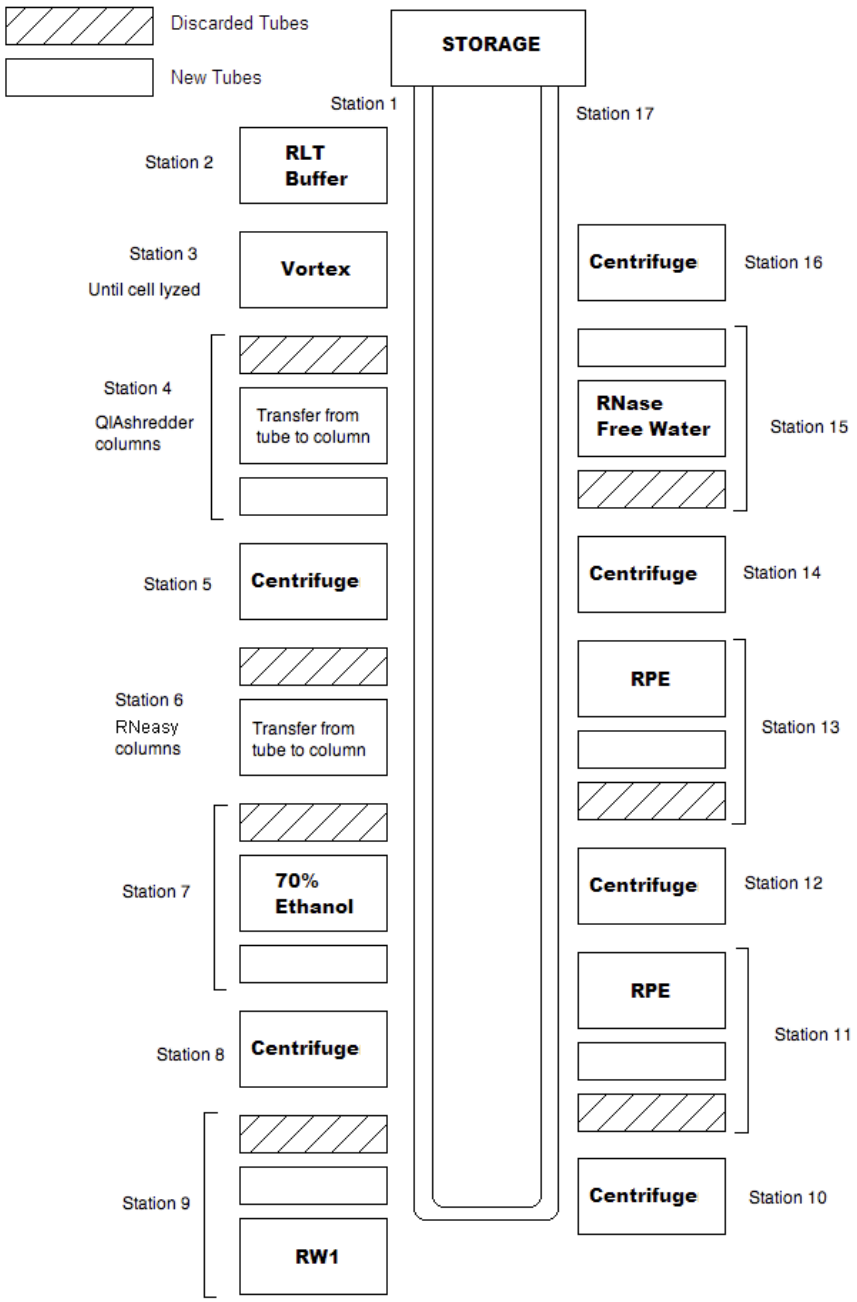

Figure 2.2 Conceptual design for the Preparation of RNA from tissue cultured cells production line

Initially the manual production line is thought of as a set of 17 stages each with one human operator in charge. That is the human operators perform simple tasks such as pipetting of reagents, discarding of used ware, preparing the samples to be loaded, and loading and unloading a centrifuge. A question that may be asked is whether each operator can run more

than one station. It can be done, but in order to keep production going with the same throughput, all timings and stations have to be well synchronized and one would have to include labeling steps in each station. In this study we opt to avoid including such extra steps and leave those for future directions.

2.3. Conceptualization of Automated Production Line

Once the manual production line is conceptually set, the operators are then replaced by automation equipment. A conveyor replaces the handing of the samples from one operator to the next. All equipment is set in close proximity and with access to and from a conveyor. The equipment needed for each station is listed next:

- Station 1: Feeder from the storage onto the conveyor.
- Station 2: De-caper, caper, and dispenser, storage RLT buffer
- Station 3: Vortex
- Station 4: De-caper, caper, liquid transfer equipment (pipette)
- Station 5: Centrifuge
- Station 6: De-caper, dispenser, storage for ethanol
- Station 7: Liquid handling equipment, caper
- Station 8: Centrifuge
- Station9: De-caper, caper, dispenser, storage RW1
- Station 10: Centrifuge
- Station 11: De-caper, caper, dispenser, storage RPE
- Station 12: Centrifuge
- Station 13: De-caper, caper, dispenser, storage RPR
- Station 14: Centrifuge
- Station 15: De-caper, caper, dispenser, storage RNase free water
- Station 16: Centrifuge
- Station 17: Feeder into storage

Each station has in addition interfaces to and from the conveyor belts to transport the columns and tubes from station to station. In other words, the first automation strategy to be

explored is one that involves no robotic devices at all. Cost wise, as robots or pick-and-place devices are relatively expensive and maintenance-intensive, if a possibility of doing away with robots altogether is feasible it should be very seriously considered. In this case study however one can discard this option from the start, as it is not feasible. The main reason is that centrifuges cannot be loaded and unloaded by means of feeders to and from conveyors, at least not the standard commercial centrifuges. These need to be loaded and unloaded by means of a pick and place mechanism as it has to be done from the top of the centrifuge.

This immediately introduces the need for at least one robotic arm (or other pick and place device) that could perform this task of loading and unloading of the centrifuges. As seen in the description of the protocol, this case study procedure utilizes no less than six centrifuging steps. Questions that need to be raised are: Should the automation designer attach a robot for each centrifuge station? Do we need to have six distinct centrifuges at all? The latter question can be answered right away: A reduction in the number of centrifuges indicates that the centrifuges become involved in more than one step. This means that some sample labeling steps have to be included. Once again, we choose to avoid adding extra steps to the protocol and we shall leave labeling steps as an open issue for future research.

Let us return to the first question about one robot for each centrifuge. There are of course six possible options depending on how many robots are to be employed. This is where computational tools come into play. Utilizing six robotic devices does increase significantly the overall system's footprint due to each robot's workspace and it increases the total capital cost as a robot is surely one of the more costly pieces of hardware in the system. However it could theoretically yield the best throughput as each of the robots is always ready to attend its respective centrifuge. At the other extreme, using only one robotic device decreases footprint and cost six-fold, but it introduces a timing complexity that might compromise the system's throughput.

Some tasks are critical and need to be performed as soon as samples become ready. Consider for instance the task of Station 14: As soon as the RNA becomes exposed in the column a sufficient amount of RNase free water must be added in order to get the largest amount

of RNA. Clearly, one of the solution outcomes of this multi-objective automation design problem (which involves equipment cost, floor footprint and throughput) should be a recommendation as to how many robots need to be used in the automation implementation of specific protocols.

On the other hand, centrifuging cannot be performed with only one tube or column in; centrifuges must be geometrically balanced in order to work properly. That means that depending on the number of available slots that a certain centrifuge chamber may have, a certain number of samples could potentially be loaded. If for example the centrifuge has six slots, it could only simultaneously process two, three, four, or six samples symmetrically arranged. Bigger centrifuges have more potential combinations. In High-Throughput automation the process involves microarrays (rather than single tubes) and centrifuging of such arrays is often done in pairs as two arrays occupy symmetrically the centrifuge slots space. Therefore the important consideration that has to be kept in mind is that in centrifuging stations more than one sample has to be loaded prior to each centrifuging action, and how many samples are to be loaded is one of the design parameters. This is a timing calculation and synchronization problem that has to be answered as part of automation design. Some of the centrifuging steps take relatively long time, significantly more than other steps in the RNA isolation process.

One thus begins to appreciate the complexity of the automation design problem.

3. BIOTECHNOLOGY AUTOMATION LITERATURE REVIEW

3.1. Introduction

The previous chapter presents a motivating example to help illustrate possible issues that have to be taken into consideration whenever designing automated solutions for biotechnology protocols. These unique features are very much related to the intrinsic characteristics of the raw materials and the products, both are biological, the processes that needed to handle such biological substances, and typical needs that are encountered in the field of biotechnology. Examples to these unique characteristics are:

- Limited availability of raw material
- Constrained lifetime of the raw material and the intermediate products created along the process
- Specific environmental conditions needed for the raw material (and intermediate products) and the processes to take place.

Other automation design concerns illustrated in the previous chapter are common to all fields of Industrial Engineering such as:

- Timing constraints
- Sample manipulation architectures
- Queues
- Equipment sharing
- Total throughput
- Economically reasonable solutions

This chapter reviews existing biotechnology automation solutions. Examples developed, both in industry and academia, are discussed in part with reference to questions posed in the previous chapter.

3.2. System Level Approach to Flexible Automation for Biotechnology

Najmabadi et al (Najmabadi, 2003, 2005, 2006abc) advocated a system level approach to the study of high-throughput automation for both upstream and downstream protocols in small-to-medium size biotechnology laboratory systems. His study concentrated on small-to-medium size laboratories as their number is considerable, and as such a research of such systems was deemed to deserve special attention (Najmabadi, 2006a)

In (Najmabadi, 2003) a conceptual design for an automated high-throughput magnetic bead protein complex purification work cell was demonstrated with a special focus on the Tandem Affinity-Purification (TAP) protocol, considered a major Proteomics field protocol used whenever studying protein-protein interaction. The TAP protocol is based on the fusion of a specific TAP tag to a target protein complex in order to later perform an affinity isolation of the fusion protein using bead chromatography columns. The protocol in its entirety consists of seven steps. In (Najmabadi, 2003), only the sixth step was automated in a conceptual design. That step consisted of a magnetic bead purification chosen for Protein Complex Purification because it tends to be a more automation friendly protocol as it does not involve traditional but hard-to-automate methods such as centrifugation and time-consuming column-draining processes. Even though this was only a theoretical and conceptual example, it clearly concurred with the challenging steps and bottlenecks for automation found in our Chapter 2 case study, such as loading of specific pieces of hardware equipment and the time taken by some processes to complete. For example, some operations such as incubation, lysing or amplification, operations used in Isolation for RNA, take a given amount of time for the reaction to be carried out properly and that service time cannot be changed. On the other hand, equipment such as conventional centrifuges as well as vortexes, off-the-shelf equipment used for Isolation of RNA, have to be loaded from the top as the tubes containing the samples have to go each into its individual slot.

The above study claimed that the tendency in laboratory automation has been to favor flexible systems, enabled by means of articulated robots or distributed motion systems, moving away from implementations based on moving conveyors only. Looking back at the case study presented in Chapter 1, right from the start one has to agree that the use of robotic devices is not

only a tendency, but it is compulsory. The protocol presented as a case study for RNA isolation has both kinds of issues mentioned in (Najmabadi, 2003) as “hard-to-automate methods of centrifugation and time-consuming column-draining processes”.

3.2.1. Flexibility Parameters

In the context of biotechnology automation, the flexibility of laboratory automation systems was defined in (Najmabadi, 2006b) as the capability of the system to adapt to a new protocol. Three main quantitative parameters to model aspects of hardware flexibility in Laboratory Automation systems were identified as: functional flexibility, structural flexibility, and throughput flexibility.

Functional flexibility is related to the transportation capabilities of the system and it is divided into three aspects: liquid handling flexibility, transportation flexibility, and instrument feeding flexibility. Liquid handling flexibility refers to the capability of the system to accurately dispense and aspirate a large range of liquid volumes. In biotechnology, there is a need to be able to dispense very accurately fluid volumes in the order of magnitude of micro-liters. In the case study presented in Chapter 2, for the manual process, the volumes used are in the order of magnitude of milliliters. One of the goals behind automation of such processes is to be able to have the reaction volumes reduced to the order of magnitude of micro-liters.

Transportation flexibility refers to the system’s capability to manipulate various kinds of labware such as vials, tubes, plates, etc. and move these successfully from point A to point B. In high-throughput automation for biotechnology the common labware involves capillaries and assay plates that can handle adequate small reaction volumes.

Instrument feeding flexibility refers to the system’s capability to load and unload such labware into different instruments and lab equipment. Conventional off the shelf equipment may be somewhat rigid in this aspect, i.e., equipment such as centrifuge can only be loaded from the top via a pick-and-place device, but nowadays customized centrifuge equipment become available and there might be more options in the market. For this reason, small-to-medium size laboratories and some big size laboratories need systems that can offer Instrument feeding flexibility to be able to integrate all types of equipment (conventional and customized).

To be able to achieve complete functional flexibility for a production line that could theoretically perform RNA Isolation, transportation flexibility and instrument feeding flexibility both need to be taken into consideration simultaneously. Both issues are very closely related to that of how the actual equipment used operates and performs the many tasks.

Structural flexibility is the system's adaptability to new instruments and protocols and it can be achieved by changing the system layout configuration or its transportation methods. The group technology issues and questions could be tackled taking advantage of structural flexibility. The idea behind group technology and classifying of different protocols into families depending on their similarities is to be able to process multiple protocols with as few and minor modifications of an already existing line. If such lines were designed keeping in mind structural flexibility, it would be easier to achieve this goal.

Throughput flexibility is the system's capability to increase its throughput by means of parallel processing solutions. As explained throughout Chapters 1 and 2, timing requirements and constraints, as well as reaction volumes and concentrations specifications tend to be tight for biotechnology protocols and samples; i.e., most operations have very restrictive specifications which have to be followed in order for the reactions to take place and the samples to stay viable. If these specifications are not followed the whole protocol may not work at all which may cause potential costly degradation damage to the samples and reagents. In addition, biotechnology protocols tend to be sequential steps, non-commutative that have to be followed to the letter (i.e.: a sample cannot be purified or homogenized before it is being lysed). These two issues clearly explain why the only practical option for increasing throughput is by means of parallel processing solutions. An in depth study of these issues is presented in Chapter 7.

The three flexibility parameters were then combined in (Najmabadi, 2006bc) to create a single hardware flexibility index. Nam Suh's Axiomatic Design Theory (Suh, 2001) is applied in (Najmabadi, 2006bc) to measure and quantitatively compare Hardware Flexibility indices using the parameters described above. An evaluation of hardware flexibility for robotic-based and track-based approaches, considered conventional laboratory automation approaches, is presented in

(Najmabadi, 2006bc). At this point in time, the design considerations for biotechnology protocols presented in this dissertation, do not involve quantitative flexibility considerations.

3.2.2. Laboratory Automation Depending on Labware Manipulation

There are two typical and widely used automation approaches whenever it comes to labware manipulation: robotic-based approach and track-based approach.

In a robotic-based approach a robotic arm moves various labware among different laboratory instruments. Robotic-based systems typically consist of a tabletop platform on which various laboratory modules and instruments are mounted all serviced by a single centrally located robotic arm or by multiple robot manipulators. The robots perform the necessary liquid handling and labware transportation operations. A track-based system on the other hand uses conveyors, rather than robots, as the transportation means between stations. In such systems modules and instruments are positioned next to the transportation tracks and the containers are moved between stations on the conveyor tracks. Each station may use feeders to retrieve tubes or other containers from its local storage.

Two architectures presented in (Najmabadi, 2006ab) termed Total Modular Laboratory Automation (TMLA) and Distributed Operations Laboratory Automation (DOLA), were introduced as means for enhanced hardware flexibility of laboratory automation systems.

The TMLA was introduced to overcome some of the shortcomings of the robotic-based approach. It was proposed to substitute the robotic arms in robotic-based approach architectures by modular and reconfigurable arms. By replacing conventional robotic arms by modular reconfigurable one, such can be placed in such a configuration as to increase the workspace of SCARA modular arms for pick and place lab operations or a as change to an articulated configuration suitable for stacking operations. In (Najmabado, 2006c) it was demonstrated that TMLA improves structural and throughput flexibility of conventional robotic-based approaches.

The second improvement method, named DOLA, is based on deploying transportation rails network which is installed above the tabletop platform on which the instruments are placed. This is done in order to conveniently move samples to all the necessary manipulator grippers and pipettors that perform the labware manipulation and liquid handling operations respectively. By

adding such transportation rails network actual dispensing stations can be avoided and, depending on the grippers used, some pick and place devices could be removed from the line as well. This network would allow easier feeding solutions of the various laboratory equipment. DOLA was claimed to improve transportation and instrument feeding flexibility of conventional track-based automation approach (Najmabado, 2006)

TMLA and DOLA architectures were conceived to improve flexibility of conventional architecture approaches in order to perform labware manipulation and liquid handling operations for tabletop systems. Even thought, they were conceived for tabletop operations they could be scale up to larger solutions with help of customized equipment and integrated software management systems.

The above two architectures will not be used in this dissertation. The implementation of TMLA architecture involves customized manipulators, and its development and acquisition may be very expensive and time consuming. The implementation of DOLA architecture involves the removing of redundant stations and the putting of more emphasis on equipment sharing. Such equipment sharing necessarily involves introduction of labeling steps that as was already mentioned is an operation not considered in this study and is thus left for future research.

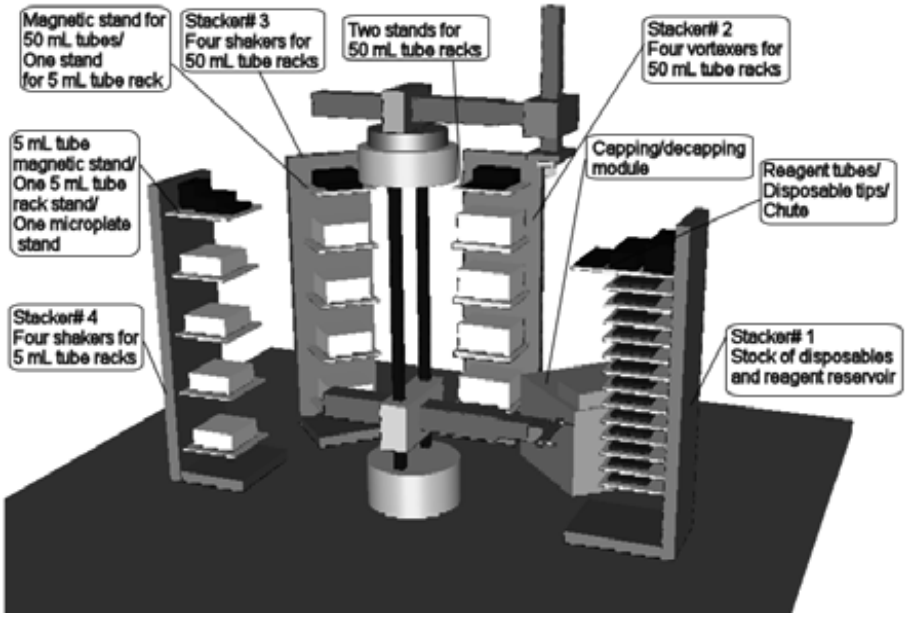

Figure 3.1 Tower-based automated work-cell conceptual design for magnetic protein complex purification protocol (Najmabadi, 2005, 2006ab).

A third architecture described in (Najmabadi, 2005) called Tower-based architecture is claimed to improve attaining the main requirements in small-to-medium biotechnology labs for batch protocols (Figure 3.1) such as throughput to footprint ratio. This architecture consists of a single central robotic tower where two cylindrical robotic arms are mounted. The instruments are placed around the central tower in multilevel stackers such that the top robotic arm handles instruments situated at the top levels of the stacks and the bottom robotic arm handles the equipment situated in the bottom levels. The bottom cylindrical arm is responsible for the pick-and-place and labware manipulation operations and the top cylindrical arm deals with the pipetting and liquid handling operations.

For sequential protocols, as most of the biotechnology protocols happen to be, having two manipulators: one for handling manipulation and the other for liquid handling operations helps to significantly reduce most of the queues and bottlenecks. In addition the layout footprint is reduced as the dispensing unit becomes the same for all liquid handling operations. The issue is that this kind of dispenser is still a customized solution and the fact that the manipulators work at two different levels makes this a 3-D solution. In this dissertation we focus only on off-the-shelf equipment.

As explained in Chapter 1, our chief objective in this dissertation is to develop general guidelines for the design of fully automated systems for upstream biotechnology protocols.

One of the key contributions of (Najmabadi, 2006) was the observation that the design of biotechnology automation systems revolves around automation of generic operations and processes termed Laboratory Unit Operations (LUO). These LUOs are the building blocks for most, if not all, known biotechnology laboratory protocols.

3.3. Customized Automated Solutions

3.3.1. ACAPELLA 5K – Liquid Handling System

Notable academic research on the subject of Automation Design for biotechnology was conducted at the University of Washington. The result of a ten years development work (1995 – 2005) was the various versions of the ACAPELLA system (Daoura, 1999) (Meldrum, 1994, 1995, 1997, 1999, 2000ab, 2001ab, 2003) an automated liquid handling system developed for sample

preparation. The system takes DNA (or other biochemical samples) and processes it as specified in a user-defined protocol. Readily available processing operations include aspiration, dispensing, mixing, thermal cycling, and imaging all performed in a capillary format. The instrumentation of ACAPELLA includes a rotary table, glass capillaries, low-volume disposable containers, and piezoelectric dispensers for precision delivery of reagents (Figure 3.2). The system successfully demonstrates high-throughput preparation of restricted enzyme digestion, PCR, and sequencing reaction for genome analysis.

The ACAPELLA system was developed with three main objectives in mind that at the time were considered to be crucial issues for future developments and advances in biotechnology as related to the Human Genome Project. These three main goals were: 1) the development of an automated liquid handling system that would improve throughput capabilities, 2) reduction of typical DNA sample volumes needed for PCR operations, which leads to a reductions in the reactants size and therefore should be reflected in a reduction of reagent cost, and 3) development of a closed sample-processing pipeline to make the process more efficient in order to start moving towards fully automated systems. The designers of the system were able to meet these goals. The core technology of ACAPELLA-5K could process up to 5000 samples in an eight hours shift, it could prepare final reaction volumes of 1-2µl a five-fold reduction over 2003-era volumes, and this was accomplished thanks to the system pipelined sample processing methodology (Meldrum 2001ab, 2003).

One of the key components in the ACAPELLA system layout was a rotary table (or turntable) with 18 discrete places called “chucks” (Figure 3.2), which moved the capillaries between the different stations. A capillary dispenser (Item A in Figure 3.2) could hold 5000 capillaries and was a rotor-based hopper design that dispensed a capillary to the empty chunk in front of it. The turntable continuously rotated with a period of 5.76 seconds simultaneously moving the capillaries each to its next station. In order to introduce the sample into the capillary, a robotic aspirator-gripper took the horizontal capillary from the chunk, rotated it 90 degrees to a vertical position, moved it to a microplate to aspirate the DNA sample, then rotated it back to a horizontal position and finally put it back in the chunk where it was taken from (Item B in Figure

3.2). An imaging monitoring step was performed to assure that the DNA sample was aspirated correctly (Item C in Figure 3.2). The ACAPELLA system had 8 piezoelectric reagent dispensers that could dispense volumes ranging 40-100pl (Item D in Figure 3.2). A second imaging step was included to verify the correct fluid volume (Item E in Figure 3.2). Three stations were available for mixing the volumes by means of a piezoelectric actuator that moved the volume back and forth by air volume displacement (Item F in Figure 3.2). The capillaries were then picked up by a robotic vacuum gripper and placed on to a cassette holder (Item G in Figure 3.2) (Meldrum, 2000a, 2001ab, 2003).

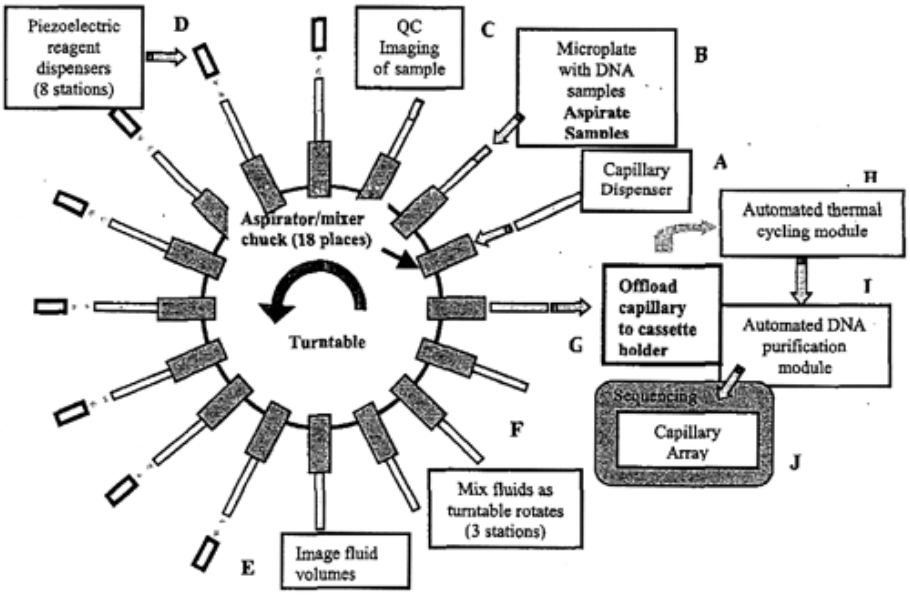

Figure 3.2 Functional schematic of fluid sample handling with the ACAPELLA-5K automated system (Meldrum, 2001ab).

One of the innovative aspects and key contributions of that long term research study was the implementation of a rotary table as a center piece pipelining operations engine. Even though a rotary table may be viewed as a circular conveyor, ACAPELLA-5K explored a novel configuration to apply it to biotechnology protocols and its environment. The rotary table in the system performed not only the manipulation operations of the systems, but at the same time it also performed the liquid handling operations since the capillaries, the labware for this application, were attached to the rotary table and the reagents were dispensed directly without having to feed or load the labware into a different instrument.

That study demonstrated that there are more automation architecture options that are available, different than the robot-based and track-based approaches discussed earlier. Fully automated systems for biotechnology could potentially combine select features and components drawn from different automation approaches.

The design team of the upgraded ACAPELLA-5K system developed a real-time PCR test-bed for the analysis of up to 48 1µl to 2µl reactions in glass capillaries. That PCR test-bed was suitable for high-throughput experimentation whenever used as a downstream module for ACAPELLA-5K. The test-bed featured a laser-induced fluorescence scanner with high-sensitivity photomultiplier tubes for the detection of three spectral wavelengths. The scanner was used to monitor the reactions taking place inside the test-bed, in order to detect the copied genes (Meldrum, 2001ab).

The PCR processor developed by the ACAPELLA-5K research team, was designed with the sole purpose of fulfilling the ACAPELLA-5K implementation requirements and goals. This means that it was designed to be able to process the number of ACAPELLAs capillaries labware required in the time required, and even though this was a customized solution for a specific system, it was not fully automated. Loading of the pre-processed samples (steps in the rotary table) into the PCR was not yet automated. To fully automate this solution, at least one robotic manipulator should have been involved in order to move and load the labware from the rotary table into the PCR (Meldrum, 2003).

This last point concurs with some of the issues raised in Chapters 1 and 2. Evidently there are various lab instruments and operations that may become bottlenecks due to their timing conditions and loading/unloading mechanisms. At this point, in an intuitive manner, it can be stated that in order to design a fully automated system, the system would need at least one robotic manipulator to perform some transportation operations. The best solution may be a composition of both conventional approaches, where some manipulation operations are performed by conveyors, but some still require robotic manipulation.

The ACAPELLA system was developed keeping in mind automation flexibility issues. It was designed as a liquid handling system that could perform simultaneously multiple different

user-defined protocols. The system could load up to eight different reagents that could each serve a different sample separately, meaning that each sample could be part of a different protocol. The system had three mixing stations for the requirements of different protocols. These two characteristics gave a degree of flexibility to the platform and it could be easily reconfigured or it could prepare more than one sample at the time keeping in mind that only eight reagents can be loaded simultaneously, and that the reactions were restricted to those that had mixing times of less than 17.28 seconds.

It is important to note that only a family of protocols could be performed with the ACAPELA system; procedures that require centrifugation operations, like the protocol described in Chapter 2 as a motivating example, cannot be performed with the platform ACAPELLA 5K and therefore, are not amenable to automation with such a platform. The protocols that can be performed by this system are based on magnetic beads isolation, which would be further explained in Chapter 5. Protocols that required centrifugation steps are found in the literature to be “not automation friendly” (Meldrum, 1999) (Najmabadi, 2006a). Automation of centrifugation steps is particularly addressed in this dissertation.

Looking back to the motivating case study presented in Chapter 2, the ACAPELLA system would not be capable of implementing it for two main reasons. First, ACAPELLA is not capable of performing protocols that involve any kind of centrifugation operations. As it is a customized and already built solution, it does not have a centrifuge, or the capability to be reconfigured to include one. This system was designed and built for an already defined set of protocols, and protocols that use centrifuges are not part of that set. Second, since ACAPELLA was built to handle only capillary tubes, there are no commercial centrifuges that can operate with capillary tubes. A new system, inspired by the ACAPELLA system, that incorporates a rotary table approach with centrifuges, the means to load them and unload them (pick and place device), and a suitable labware format, could in principle perform the RNA Purification protocol as described in Chapter 2.

3.3.2. Multiparametric Platform for Analytical Screening of Living Cells

Another interesting biotechnology automation line used for academic research is at the University of Munich (Geisler, 2006). The research conducted using that line explores the concept of a multiparameteric platform. Such a platform was developed to carry out high-throughput downstream automation related to multiparametric screening assays.

For seven years the research team at the University of Munich focused on the problem of analytical monitoring of intracellular and extracellular parameters for high-throughput screening. The general tendency in biology and related fields is to move towards a more efficient use of biological materials. Multiparametric screening helps in the measuring of various parameters with the same set of assays, making the development of the assay, the screening campaign, and the use of the biological samples more efficient.

The research at the University of Munich was carried out in two stages. At the first stage sensor-based analysis tools were developed as well as imaging-based screening devices that could carry out multiparametric screening. These tools were developed as standalone customized equipment. The research focused on increasing the content of information that could be obtained during the screening experiments. The sensing was done primarily by lab-on-chip and microfluidics devices. The second stage of the research included a development of an embedded platform that included and integrated multiparametric, bioelectric, and biochemical sensors. The goal was to accomplish analytical monitoring of cellular parameters, and an automated imaging microscope for high-volume-data screening.

The second stage of the research was an automated solution and is thus of interest to this dissertation. Even though the scope of the research done at the University of Munich is different than the one intended in this study, there are some ideas that are relevant to this dissertation. The conceptual platform developed at the University of Munich utilized a topology of distributed intelligence and hardware-based synchronization. This allowed all integrated platform components to be timed and synchronized. It was claimed that the platform was conceived as a highly modular platform, which allowed it to be reconfigured depending on user requirements. It

included integration of probe-manipulation systems such as climate control, fluidic systems, and automated probe placement.

A notable feature of this the platform, is the hardware-level synchronization achieved by means of digital signal processors. As the “Integration Control Unit” runs in real-time, it controls every operation in the system and maintains synchronicity throughout the system. In this dissertation a MATLAB® simulator was developed to study timings and queues formation in the automation line. The simulator was developed assuming the existence of a control network throughout the production line that controls the equipment in the line and maintains the synchronization of the operations.

3.3.3. HTS-TRAC™

Tecan® is a Swiss-based leading global provider of laboratory instruments and solutions in biopharmaceutical, forensics, and clinical diagnosis (Tecan, 2008). Tecan® is a company that specializes in the development and production of automated workflow solutions. One of the systems developed for the Tecan® Integration Group (TIG) is called HTS-TRAC™ and it is a flexible track-based system developed for multipurpose cell based screening assays.

Figure 3.3 HTS-TRAC™ Tecan® system (Tecan, 2008)

The system was developed by Tecan® as an example of a fully automated compound screening system. The system was developed around the manipulation equipment (Figure 3.3),

designed to reach, load, and unload any equipment part of the system. The manipulation architecture employed for the system consists of two robotic manipulators mounted on a rail to increase the overall robots workspace area. This manipulation approach combined both robotic-based and track-based approaches. This is a spin off from classical track-based system as the track is used to move the manipulator (robotic arm), and not the parts and samples. As the robotic workspace is increased, it allows the manipulators to reach the equipment needed for the protocol. It also allows extra space to fit any equipment that need to be incorporated for different protocols. The line is completely modular, and therefore reconfigurable to fit almost any biotechnology protocol, even though it was originally developed for downstream protocols (compound screening) only.

The equipment of the HTS-TRAC™ system is all made of off-the-shelf Tecan® products. The robotic devices are Tecan® Autoloader CRS F3; such a manipulator has five liquid handling devices: Cybio CyBiWell, PE (Perkin Elmer®) Flexdrop, Thermo Multidrop®, Biotek Washer ELx, and Velocity11® Bravo. It has two incubators Thermo Cytomat® 6001; three readers: GE Leadseeker™, MD FLIPRTETRA, and PE Envision® reader; and it also uses ABgene® Piercer ASP50 (sealer for plates and tubes), and Velocity11® VSpin (centrifuge). It uses software Facts 5.0 for integration of the system. As all the elements of the systems and the system itself are developed and produced by the same vendor, it is easily integrated and everything is controlled from the one computer that also performs the data acquisition. The HTS-TRAC™ system uses 384-well-plates format and its reported throughput is approximately 200 microplates per 24 hours.

As this system is completely modular and reconfigurable, it could potentially be reconfigured to perform the motivating example presented in Chapter 2 for RNA Purification. Some changes need to be done to the original system, like the introduction of more centrifuges and an additional vortexer, and possibly the rearrangement of the equipment to be in the correct order specified by the protocol. The system however has the ability to be capable of loading and unloading of all the equipment needed for this protocol and it has the necessary layout space needed for it. This is an example of a truly flexible solution with functional flexibility, structural flexibility, and throughput flexibility. The only issue with this solution is its relative high cost.

3.3.4. Scripps - High-Throughput Screening for Drug Discovery

The Scripps Florida Research Institute has a facility completely dedicated to high-throughput screening. Their system is one of a few systems considered to be ultra-high-throughput screening (uHTS) systems. This system will be discussed in depth in Chapter 4 as it is referred to for calculation of typical upstream biotechnology design automation throughput specifications dictated by drug discovery needs (Hodder, 2007).

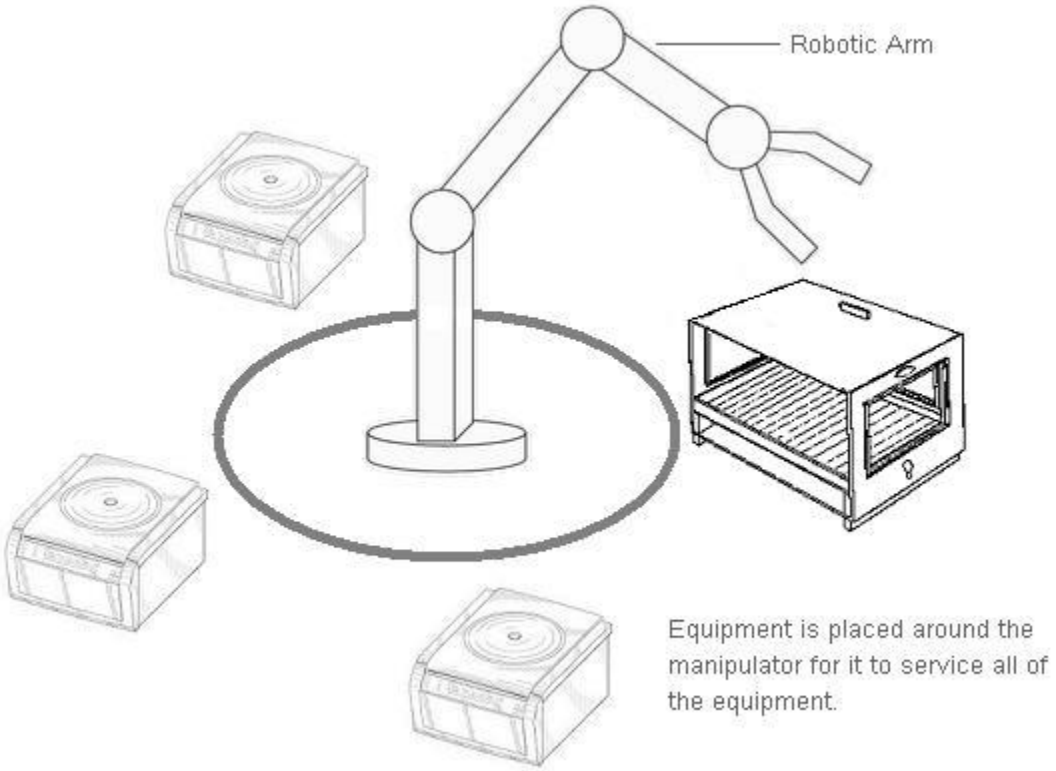

Figure 3.4 Scripps – HTS system comprehensive diagram

In this section only the automation aspects of the system are discussed. Performance issues of the Scripps system are discussed in Chapter 4. The core architecture is a typical robotic-based approach: it has a robot manipulator at the center and all the equipment is placed within the reachable workspace of the manipulator (Figure 3.4). The robotic arm performs all of the labware transportation from station to station. As the layout of the system is restricted by the workspace of the robotic arm, the introduction of structural flexibility to the system may necessitate adding extra equipment needed for implementing additional protocols. These may or

may not fit within the robot workspace. Such additional equipment may be added on top of the existing instruments resulting in a 3-D implementation solution.

The Scripps system has an interesting particular characteristic. It is programmed and controlled so that in order to reduce or avoid queues the speed of the robot is modified according to the operations timing requirements. It is a strategy for queues altogether avoidance.

The uHTS system was developed for downstream screening operations, such as compound screening and drug discovery campaigns. Nevertheless, it can be reconfigured to perform upstream protocols like RNA purification. The only issue is that it would be a 3-D solution as 17 stations (like the ones proposed for the RNA purification process) may fit around the robotic arm.

As the last two examples demonstrate, it is not difficult to find fully automated high-throughput protocols for the downstream processes. But, as ones moves to the upstream protocols and the actual harvest of the samples less and less automated equipment is found. Automated equipment for such operations just begins to emerge. The DNA Purification System and the Nucleic Acid Extraction WorkStation by Thermo and the Centro LB 960 Microplate Luminometer by Berthold Technologies are examples of this group of solutions and will be further explained in Chapter 6.

4. DRUG DISCOVERY

4.1. Introduction

This chapter presents an overview of the drug discovery process with emphasis on related process automation issues. It discusses what specific tasks within the drug discovery process can be automated, and why some of the components of the process cannot be automated. In the context of biotechnology automation, drug discovery is one of several possible “downstream” processes. Prior to initiating a drug discovery campaign, certain “upstream” biotechnology automation processes must conclude. Such processes involve samples preparation (such as a quantity of specific DNA segments or certain purified proteins), followed by assay preparation. In this chapter we explore possible throughput specifications for the upstream processes, as dictated by the needs of drug discovery (Cardona, 2012b).

In the past, most drugs have been discovered either by identifying the active ingredients in traditional remedies or serendipitously. In recent years drug discovery approaches that rely heavily on automation emerged allowing users to better understand how disease and infections are controlled at the molecular and physiological levels and how to better target specific biochemical entities based on such insight.

4.2. The Drug Discovery Process

Drug discovery is a sub-field of biotechnology involving research, design and development of new therapeutic drugs to treat known diseases. In drug discovery the focus is on understanding how diseases (including infections) work at the molecular and physiological levels in order to be able treat such diseases at the root cause of the problem.

In general, drug discovery involves the following multiple aspects: the identification of possible molecular substance candidates that need to be targeted in order to effectively treat the disease; the synthesis of plausible candidate targets; the characterization of the candidate targets and of the disease mechanism that needs to be treated; the screening of each candidate target

against a large number of compounds in order to find plausible therapeutic results; and the development of assays to identify therapeutic efficacy. Once all the above steps are performed and a compound is shown that has a possible desired effect on the target being evaluated, the process of drug development commences which includes among other factors the determining of the proper dosage of the drug and the exploration of the drug possible side effects.

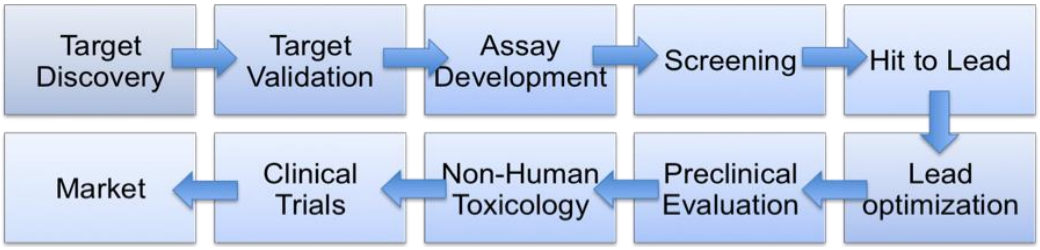

Figure 4.1 A conceptual block diagram of a complete pharmaceutical research and discovery process (Carnero, 2006)

The complete block diagram description of a pharmaceutical research and development process is presented in Figure 4.1 (Carnero, 2006). Following the exploration of the interaction of the disease with the organism at the molecular level, the target discovery process identifies target molecules that may be amenable for disease treatment. These targets may be proteins such as receptors, enzymes or ion channels (Carnero, 2006), or specific RNA or DNA segments (Hodder, 2007) that interact with the disease to enable or inhibit the progress of the illness. Based on the ultimate longer term goal of the human genome project it is safe to predict that as we may gain full understanding of all gene functions, all targets in future drug discovery processes would eventually become DNA and RNA sequences that express or inhibit diseases, totally replacing all other kinds of targets (Bartfai, 2006).

Once an appropriate target is identified, it has to be verified. A level of confidence has to be established to ensure that a given target is relevant to the disease under study, and that its manipulation may allow with high enough probability treatment of the disease. Depending on the type of target found, its manipulation (called modulation in the drug discovery jargon) is done by means of various chemical compound effectors (known as modulators). For instance, if the target was a receptor, the chemical substances used to manipulate such a target would be either

agonist, substances which combine with the receptor on the cell membrane surface to produce a desired reaction, or antagonist, substances that combine with the receptor to prevent a physiological action. If on the other hand the target found is an enzyme, the modulator used in such a case may be an activator or an inhibitor depending on the need. Finally, if the target found is an ion channel, then opener or blocker compounds are used.

Once the target is validated, the next step is an assay development where the goal is to identify the modulators mentioned above for a specific target. Assays are designed to monitor the interaction of the target with other chemical substances or compounds during the screening in order to find which substance interaction draws the desired chemical result for the target in hand.

Screening for drug discovery is a highly automated process. Since the goal is to expose the candidate target to a very large number of different chemical compounds, this process is performed using a high-speed parallel method known as High-Throughput Screening (HTS). The target has to be exposed to all those compounds in a hope of finding the desired interaction between target-and-compound. The screening process may yield more than one compound that has the desired interaction. These compounds that create successful reactions are called “hits”. Further screening needs to be run in order to find the appropriate dosages to obtain the desired result. Following the finding of the right dosage for each “hit” compound, trials are conducted in animal models. The compound that scores the best results from the animal models experimentation is then considered the “lead” compound.

Lead compounds are optimized in terms of strength and selectivity. The best compound is a one that affects only the desirable target with the desirable action, and its safety is assessed before it becomes a candidate for drug development. Some compounds might be toxic and even lethal to humans if the right dosage is not administrated.

Following the lead compounds identification and approval as candidates for drug development, drug discovery enters a phase at which preclinical evaluation and non-human toxicology tests are performed, followed by the clinical trials (Drews, 2000).

4.3. Essential Components of a Drug Discovery System

The drug discovery process system has four main components. The essential core components are compound collections (or libraries), assay designs, HTS technology platforms, and data capture/analysis systems.

Compound collections are stored in custom-designed, fully automated warehouses where the compounds are formatted and made available to be instantly retrieved for testing. These compound libraries are under constant quality control. Compound libraries are constructed from a variety of different sources. These could be natural product samples, acquired from external sources, or it could be developed from combining different compounds to form “combinatory libraries”, or these could have been developed and/or discovered in-house (i.e., by the HTS facility itself during various target discovery campaigns). Compound libraries are considered major assets within a company’s set of discoveries, as these require a significant time and effort to assemble and a big financial investment (Banks, 2007). The compounds discovered in-house have a particularly special value.

The second essential component of the drug discovery process is the assay design. It is vital for the process for arrays that are chemically relevant and pharmacologically accurate are designed to have a successful outcome in a target screening campaign. The small molecule domain is thought to have somewhere between 10^{40} to 10^{100} possible compounds, more than there ever be made (Eckstein, 2006). Usually a screening campaign tests from 10^{5} to 10^{7} compounds (Carnero, 2006). The development of the assays requires expert scientific knowledge in order to make good judgment decisions as to how to choose a sensible screening sample of compounds (i.e., make an initial estimate of what compounds or family of compounds are relevant to the target in hand). This–helps to enhance the quality and diversity of the collection being screened, and to improve the success rate of the HTS campaign (Seethala, 2008).

Designing a bioassay for a given target that is appropriate for HTS is very time consuming and is considered to be one of the bottlenecks in the drug discovery process (Banks, 2007). Some of the parameters that have to be taken into consideration in the design of an assay can include choice of assay mode (agonistic or antagonistic), availability of standard protocols

(readily available assays), reagent availability, and obviously, regarding HTS, scalability. Biotechnology automation, in general, is divided into upstream and downstream processes. For instance, upstream protocols govern the production quantity and quality of the targets required for the development of the assays for an HTS campaign. The design of such assays appears to be one of the central challenges of upstream processing and it involves no automation.

A HTS campaign needs both, an assay and a target. The target and its fidelity and quality are essential to the process. Ensuring that for all those hundreds and thousands of test runs the target is exactly the same is crucial for a successful campaign. Preparing the target sample is one of the processes needed for drug discovery that is not yet fully automated. This dissertation centers on this issues.

Equally important are assays because they will ensure the success of the campaign. The assays are not designed using automated methods. A process called assay design is involve. The assay design process is developed by chemistry and pharmaceutical experts. They must develop an insight as to what compounds might be needed based on their knowledge of the target. They can make educated guesses of what interactions are needed and what kinds of compounds are more likely to get the desired result. As a result, a good starting point for the screening campaign is available. Down the road, the assay might need to be modified and iterated to find the best result. As the assay design process involves manual labor assisted by automated equipment and the process needs constant human intervention it is consider a bottleneck for automation.

Following assay design, it needs to be replicated to have sufficient number of plates available for the drug discovery campaigns. This process is called assay preparation and its automation has been resolved. Assay preparation involves low-volume liquid-handling mechanisms to replicate a “mother” plate created at the assay design into “daughter” plates used for the screening process. Usually, by using disposable tips, small volumes are taken from the “mother” plate to dispense nanoliter volumes into “daughter” plates. These mechanisms are highly automated and are able to replicate one column or the complete plate at the time. The assay preparation process usually includes a visual inspection station to ensure that require

volumes are present in each well. The final reaction results are too measured by means of optical methods.

HTS is the third essential, and considered by some scientists to be the most important, component of the drug discovery process (Banks, 2007). Over the last 20 years biotechnology and pharmaceutical companies have spent billions of dollars on developing automated equipment for HTS. Nowadays, there is a full range of automated equipment available for HTS from the comparatively simpler like modular workstations that can perform plate-to-plate transfer (Figure 4.2), to the more complex, involving integration of equipment and robotic manipulators, like the Scripps Research uHTS platform described in a later section. Sample batch production to serve the needs of HTS design tasks necessitates the setting of quantitative specifications for sample preparation automation, the topic of this dissertation.

Figure 4.2 Labsystem MultiDrop 384 from Beckman Coulter (Beckman Coulter, 2011)

The last major component of a drug discovery process is the data management operation. There are diverse software packages that have been developed; some are commercially available and others are customized. These software packages help to capture millions of individual data points that are produced by each screening run visual interfaces, to perform detailed analysis, and to interpret the information acquired.

4.4. High-Throughput Screening

The drug discovery industry has been moving towards largely automated processes that allow the screening of millions of compounds against various biological targets. HTS is defined to be a process by which hundreds of thousands to millions of compounds are tested for activity

against disease targets of interest with the goal of identifying truly active evolutionary “hits” Banks, 2007). The advance and evolution of HT platforms to be integrated into the drug discovery process are mainly classified into five phases.

Figure 4.3 FlexStation 3: High efficiency dual monochromator-based optical system for all read modes from Cisbio (CISBIO, 2011)

The first phase is the evolution of instruments and screening formats. The drug discovery community had to come to a consensus as to which formats to use in order for assays, consumables, and equipment to be compatible and integrated. Modular semi-automated and automated workstations have been developed in accordance with those formats. Plate-to-plate transfer, reagent dispensing, centrifuging, and detection capabilities (Figure 4.3) are among the operations that have been automated to use HT formats.

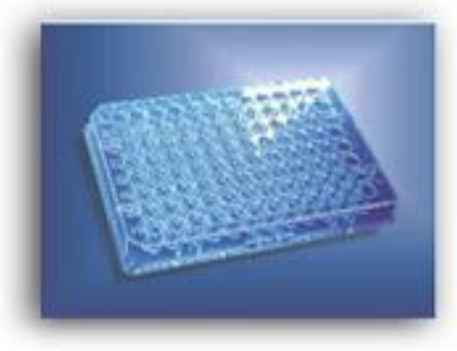

Figure 4.4 Immulux™ 96-well microplate form Dynex (DYNEX, 2011)

A second evolution phase is miniaturization. Standard format 96-well plates (Figure 4.4) and 384-well plates (developed in 1996) can handle volumes of 50µl per well. Later on 1536-well plates were developed with volumes of 5 to 10µl per well (standardized in 2003). This last

screening format is sometimes referred to as Ultra (or micro) High-Throughput Screening or uHTS. Customized formats like the 864-well plates from Affymax, 9600-well plates from Dupont Pharmaceutical, and 3456-well plates from Aurora Bioscience have been developed as well. This last one has become standard format for uHTS as miniaturization continues to evolve.

A third phase is the development of cellular HTS assays. The assays have to be compatible with the format used in HTS. The developments of this and the above two phases are of course closely related. If for example, the format is miniaturized, the instrumentation and assays have to be miniaturized as well, which translates into, for example, new dispensing techniques with lower volumes and greater resolution accuracy.

A fourth phase is the actual development of the HTS technologies. Compound selection and quality control are big issues, but screening campaigns have to include tests for drug-to-drug interaction, drug metabolism and absorption and permeability issues that the target could present.

The fifth phase is the quality control of HTS processes. The information collected from a screening run is very closely related to the supply of robust reagents, and this is becoming an integrated part of the process. Assay quality, compound quality, and target quality are constantly monitored. The success of the campaign depends, in part, on the quality of the products used. Consequently, some quality control processes deals with dispensing the compounds into assays, storing compounds, guaranteeing the purity and stability of the compounds, developing detection devices to measure assay products, and capturing and analyzing data.

4.5. Example: HTS for Drug Discovery at Scripps Florida

A High-Throughput Screening platform has been developed at The Scripps Research Institute® Florida. It enables drug-target lead identification via ultra-high-throughput screening (uHTS) technology (Figure 4.5) (Hodder, 2007). Using state-of-the-art instrumentation techniques used for drug discovery such as HTS Assay Development, HTS Target “Profiling” Assays, and Full-Scale HTS Screening, can be executed. HTS assays have been developed in microliter plates having 96, 384, and 1536 well formats.

The main component of the Scripps uHTS platform, a HT Kalypsys robotic system, is an anthropomorphic robotic arm (Item A in Figure 4.5) that moves assays and compound microtiter

plates among laboratory instruments for processing. The remaining equipment is laid out around the robotic arm within its reachable workspace to allow convenient transfer and manipulation. A pin tool is (Item B in Figure 4.5), is used to transfer compounds from the compounds plates to the assay plates. The pin tool has the same number of pins as the number of wells in the plate; i.e., if the plates that are being used are of a 384-well format then the pin tool must have 384 pins, one for each well to allow transfer of the complete plate contents all at once. The uHTS platform also utilizes an integrated liquid handler capable of dispensing up to 32 different reagents as well as of washing the plates as needed (Item C in Figure 4.5). Incubators are laid out around the arm (Item D in Figure 4.5). These have a capacity of storing up to 700,000 samples in 1536-well format at different temperature and gas concentration levels. A multimode plate reader performs measurement operations (Item E in Figure 4.5). It records absorbance, luminescence, fluorescence, or fluorescence resonance energy transfer (with time resolution) information from microtiter plates. Another measuring device is a kinetic imaging plate reader that allows measurement of second-messenger or ion channel activity in live cells (Item F in Figure 4.5). The uHTS system needs is another robotic platform to manage and distribute the more than 600,000 compounds used for drug discovery Scripps Research. The Scripps automation HT robotic system has a throughput of over 1 million assay wells per day and an on-line storage capacity of over a million compounds for both biochemical and cell-based drug screens.

Figure 4.5 The Scripps Research uHTS platform. a) Industrial anthropomorphic robotic arm, b) pin tool, c) liquid handlers, d) incubators, e) multimode plate reader, and f) kinetic imaging plate reader (Hodder, 2007)

The Scripps uHTS system is almost an exception in the current drug discovery market. There are not that many laboratories that have such processing capabilities. This uHTS system is an example of a fully automated system that is quiet costly.

4.6. HTS for Drug Discovery: Specifications for Upstream Biotechnology Automation Processes – Example Based on the HTS System at Scripps

Not every biotechnology laboratory that conducts research in target identification has a High-Throughput Screening facility at its disposal. Much smaller scale target identification research is conducted in the academia or in small-to-medium size laboratories. Such laboratories sometimes rely on larger facilities, such as Scripps, to run their screening campaigns. The HTS facility needs the following information in order to run the campaign: target name, target class, pharmacology probed, assay type, current detection technology, protocol format, current assay format, and whether or not positive and negative controls are available.

As mentioned earlier the standard formats for HTS and uHTS are 384-well, 1536-well and 3456-well plates. The first involves fluid volumes of 50µl, the second 10µl (Eckstein, 2006), and the later, involving plates that are sometimes referred to as nano-plates, uses volumes in the order of magnitude of hundreds of nano-liters (Chung, 2008).

As mentioned earlier the current trend of many drug discovery campaigns is to run screening of approximately one million compounds. As a “rule of thumb” a rate of success that has been experienced many times is that for every 500 compounds screened, one “hit” is typically found. For every two hits found, one is likely to become an “active hit”. For every 20 active hits that are screened, one “lead” may eventually be found. For every five leads found, one may become a “candidate”. Finally, for every 10 possible candidates, only one could become a marketable drug (Carnero, 2006).

An HTS campaign involves more stages and procedures than just the screening of targets against the Compound Library. An HTS facility is the one that develops appropriate assays for the HTS campaign. Assay development is done first in 96-well format plates, and once the right volumetric ratio between substrate, target, and measurement reagent is found, the assay is miniaturized to one of the desired HTS formats. To ensure that the assay functions properly,

some campaigns prepare 15 copies of each plate to be run. This means that if an assay is developed then in the first try at least, just for the sake of assay development the amount of target needs to be:

$$VP = VTW \cdot NW \tag{4.1}$$

Where *VP* is the volume of the target needed for each plate, *VTW* is the volume of the target needed per well, and *NW* is the number of wells in a plate. Therefore the fluid volume needed for the initial assay development in a 96-well plate format would be 50µl × 96 = 4.80ml per plate. As 15 plates are run to validate the results the actual volume needed for the initial assay development is 4.80ml × 15 plates = 72.00ml.

By (4.1) the volume for the miniaturization process of the assay can be estimated as well. For a 384-well format only columns 3 to 22 are used for this purpose totaling only 320 wells. Columns 1 and 24 are not used at all in order to avoid edge effects and columns 2 and 23 are used for control/quality purposes (Hodder, 2007). This means that the volume needed for assay development at the 384-well format is 24µl × 320 = 7.68ml per plate. The volume needed for 15 plates would then be 7.68ml × 15plates = 115.00ml.

For a 536-well format 1280 wells are used taken from columns 5 to 44. The volume needed per plate is 1µl × 1280 = 1.28ml and for 15 plates the volume needed is 19.20ml.

For a 3456-well format 2880 wells are used in columns 7 to 66. The volume needed for this format per plate is 0.7µl × 2880 = 2.02ml and for 15 plates 30.02ml.

The total volume needed for assay development is calculated by adding the different volumes calculated above needed for assay development. Thus 237ml of target is needed for the assay development process when miniaturized to the smallest format (3456-well plates). This is only an amount calculated for the volumes used in (Chung, 2008). The total volume needed for any other assay development is expected to be of the same order of magnitude. Practitioners often recommend the doubling of the above assay development calculated amount Sometimes the target assays are not miniaturized to the smallest HTS formats. There are occasions that these volumes cannot be miniaturized due to molecular and biological properties, however the target volume should be kept the same for any unexpected complications.

Following the development of the screening assay, target screening against the compound libraries is performed. Primary hits (active compounds) are typically confirmed using 384-well format and targets are counter-screened to eliminate false-positive results from the primary hits. This cycle is often repeated three times in order to produce three replicate results. The compound library screening and counter-screening may be performed using either 1536-well format or 3456 well format; yielding a total screening capability of 300,000 or 600,000 compounds libraries, or even 1 million compounds library.

Table 4.1 Volumes needed for target screening against the Compound Libraries

Format	Library size (number of screened compound)	Volume per well	Total volume in Liters (l)
1536-wells	1,000,000	10µl	10.00
1536-wells	600,000	10µl	6.00
1536-wells	300,000	10µl	3.00
3456-wells	1,000,000	0.5µl	0.50
3456-wells	600,000	0.5µl	0.30
3456-wells	300,000	0.5µl	0.15

The Total volumes shown in Table 4.1 in liters reveal what is needed for a single campaign run. Often as mentioned in (Chung, 2008) at least three replicate results are obtained to assure good results, and this is done both for the screening and the counter-screening. Therefore at least twice of the total volumes shown in Table 4.1 is needed in this stage for each campaign, but as much as six times the volume shown in Table 4.1 can sometimes be used for this stage. The total volumes in Table 4.1 were calculated using (4.2) by multiplying the volume of target needed for each well times the number of compounds that the target will be run against.

$$VSC = VTW \cdot NSC \tag{4.2}$$

Where *VSC* in the volume of target needed to screen the number of compounds in the chosen library, *VTW* is the volume of target needed per well, and *NSC* is the number of compounds that are screened. In order to run a HTS campaign of one million compounds with no replicated results, the targets volume that is needed could be as large as 20 litres (Table 4.1). Whenever a HTS laboratory runs a HTS campaign for a client it is a common practice that the amount requested for the screening is increased by 50% to provide a comfortable safety margin to develop the assays for the screening campaign.

An appropriate amount or volume for a target is equally important to the concentration of the target in the sample. The concentration needed for the assays are in the order of magnitude of 0.1 to 0.05mM (Drews, 2000). Both, quantity and concentration are to be specified for the purification of the starting target. The user has the liberty of choosing the appropriate protocol for the upstream (sample preparation) process taking into account these two features.

Any target shelf time may have limits depending on the biological and molecular properties of the target. Very often these might have a shelf time that may be longer than what an entire HTS campaign may last. However HTS facility may prepare fresh reagents for the campaign every 8 hours, as done for instance in (Chung, 2008), to guarantee a successful campaign. Very often, in order to ensure the quality and freshness of the targets, the targets are prepared in at least two batches, one that should be provided at the beginning of the campaign and the other one that commences near the middle of the campaign process.

As any solution including targets can be diluted from their original concentration but the inverse process is not possible, the target needs to be purified at the highest concentration specified, and a higher concentration is often recommended. It is a common practice to try to prepare samples at the maximum concentration of 1M but this is not always possible for many samples (Chung, 2008).

4.7. Role of PCR in Upstream Processes

Normally the total amount of the target is not prepared straight from a biological source. Amplification methods are often used on those samples obtained from biological sources in order to meet the quantity requirements. Reliance on samples preparations from biological sources only is totally impractical as the preparation process could take many months. Amplification methods create duplicate copies of the samples at a fraction of such time. Depending on the type of target, RNA, DNA or Protein, the user has many options with regard to the choice of duplication rates. It is up to the user to select the appropriate amplification methods. For example, if the target sample is a DNA segment, Polymerase Chain Reaction (PCR) must be the method of choice (Ngatchou, 2006).

PCR consists of three main steps: denaturing, annealing and elongation, and these are carried out at different temperatures (Florida Atlantic University, 2006). Using a PCR machine (thermo-cycler), the sample goes through the different temperature changes. First at a temperature that usually runs between 94° – 98° C, DNA is denatured. The temperature is then lowered to the range of 50° – 65° C for the primers to attach to the DNA templates (annealing), and last, the temperature is increased to the range of 75° – 80° C where elongation takes place. In the elongation step DNA polymerase synthesizes the complementary DNA strand to the DNA template by adding dNTPs to the strand. DNA target is thus doubled. Repeating the process hence leads leading to an exponential amplification. Finally, the cycle starts again, and there are as many cycles as required or for as long as there is enough reagents material to synthesize the new strands. Each cycle could take up to 5 minutes but this depends on the size of the DNA segment that is being copied. There is also an initializing step and a final hold step and each can take up to 10 minutes. Common PCR methods can only amplify DNA segments of up to 10 kilo-base-pairs. After the first exponential amplification stage, PCR enters a leveling off stage where the reaction slows down as the enzyme loses activity and the reagents are consumed. At the end the reaction stops altogether due to the exhaustion of reagents and enzyme whichever comes first. It is a common practice to run no more than 40 cycles, meaning that one original DNA segment could be amplified during 14 hours to 1.0995×10^{12} copies. PCR reduces drastically the time taken to prepare the required samples for drug discovery and it can be used to increase the samples concentration.

Protein Misfolding Cyclic Amplification (PMCA) is a protein amplification technique, one of many that exist, that is conceptually similar to PCR, and for this reason it has the same exponential growth of amplifying segments as that of PCR. In the case of protein amplification the reagents become inactive more rapidly and therefore whenever running the protocol it is a common practice to run it for no more than 20 cycles, meaning that one original protein molecule could be amplified in 8 hours to 1.0485×10^{6} copies (Amersham Pharmacia Biotech , 2011) (Soto, 2002). RNA Amplification is a technique that has been recently developed (NuGEN, 2011) that is

faster than other techniques mentioned above. The protocol can be performed in a day and it also has an exponential growth (Ngatchou, 2006).

Finally, it is important to mention that depending on the size of the HTS laboratory, the number of campaign runs per HTS unit tends to be between 10 to 30 runs per year; each takes about 12 to 36 days. Campaigns not always result in success. Drug development processes are sometimes abandoned due to negative results.

5. UPSTREAM BIOTECHNOLOGY PROTOCOLS

The biotechnology protocols presented in this chapter serve to expand the case study of Chapter 2 in order to explore various automation approaches and their varied characteristics. These protocols are merely a representative subset of upstream biotechnology procedures reported in the literature.

The protocols were chosen from a large collection of processes considering three main features: the experience with the manual protocol to make its extrapolation to the automation environment possible, the effectiveness of the protocols within the sample preparation for drug discovery context, and the helpfulness of the protocols to demonstrate and explain the various automation issues presented in this dissertation (QIAGEN 1997, 2001ab, 2002, 2003, 2006ab, 2007ab, 2008ab, 2012).

5.1. Purification of Total RNA

The complete protocol for purification of total RNA is divided into three main processes: 1) disruption of the input sample, 2) homogenization of the sample, and 3) isolation of RNA (Figure 5.1). In general, biological samples are first lysed and homogenized in the presence of a highly denaturing buffer, which immediately deactivates RNases to guarantee a purification of undamaged RNA. The actual isolation of RNA is carried out using column chromatography, a well-established technology for RNA purification. It combines the selective binding properties of a silica-based membrane with the speed of spin technology. For column chromatography, ethanol is added to provide appropriate binding conditions so that RNA in the sample can attach to the column's membrane. The sample is then transferred to a chromatography column, where the total RNA binds to the membrane, and other particles and molecules in the lysate are washed away. At the end of this process, total RNA is eluted from the membrane.

Disruption and homogenization of the starting material are two distinct steps and both are vital for RNA purification. Some methods perform both steps: disruption and homogenization of the sample.

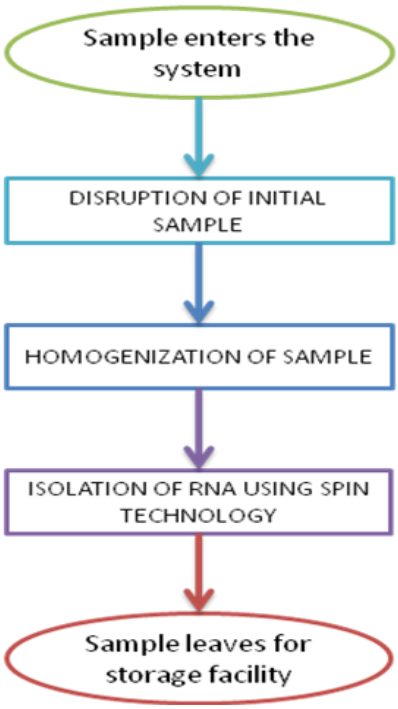

Figure 5.1 Process flow chart for Purification of RNA using spin technology

As RNA is found (depending on the type of RNA) in the nucleus of the cell, in its cytoplasm, and attached to its ribosomes, complete disruption of cell walls and plasma membranes of cells is critical in order to release as much as possible of the RNA contained within the samples. Different starting samples require different disruption methods. If cell walls and plasma membranes are not completely broken down, RNA production yield may be reduced since some RNA trapped inside those cells may not be isolated.

Homogenization is required in order to reduce the viscosity of the lysates produced by disruption. Homogenization shears genomic DNA and other cellular particles to create a uniform lysate. If homogenization is incomplete, RNA may not bind effectively to the chromatography column, and therefore, RNA purification yield may suffer.

The isolation procedure of RNA is conceptually the same for all types of raw samples. Some of the final steps may vary according to yield and concentration requirements but not according to the type of the starting raw material.

Sections 5.1.1 to 5.1.4 describe the similarities and differences among the protocols used for RNA purification for animal cells, plant cells and animal tissues.

5.1.1. Purification of Total RNA from Animal Cells

Purification of Total RNA from animal calls has already been described step-by-step in Chapter 2. In this section we only focus on the generic issues regarding the operations.

The disruption method used in this protocol for animal cells is the addition of lysis buffer, and the homogenization method is performed by means of a homogenization column. All binding, washing, and elution steps are performed by centrifugation in a microcentrifuge.

Disruption Method

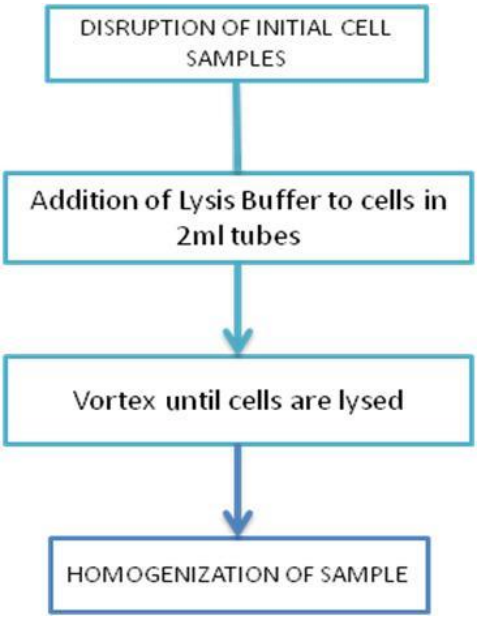

Figure 5.2 Disruption method for animal cells: lysis buffer and vortex

Disruption of raw cell samples consists of lysing of the cell membranes by adding a lysis buffer to the sample originally contained in a 2ml tube and by the vortexing or shaking of the tube until cells are lysed (Figure 5.2). At this point in the process, the sample in the test tube goes from a clear transparent color to a milky one. For a scientist performing this protocol manually, this is how the success of the lysing operation is determined. In automation, computer vision may be

used to detect the color change. However yield-improvement feedback control is beyond the scope of this dissertation.

Homogenization method

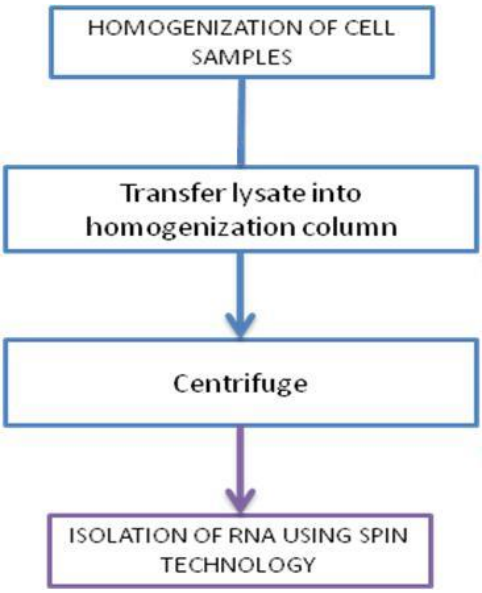

Figure 5.3 Homogenization method for animal cells: homogenization column and spin technology

The homogenization method for animal cells consists of two steps: first, the lysate is transferred to a homogenization column, thereafter, the column that is attached to a 2ml collection tube is spun for a certain period of time to force the entire contents through the column's membrane, which in this case shreds the bigger molecules and particles in the lysate (Figure 5.3). The result is a uniform sample obtained in a 2ml collection tube used for further isolation of RNA.

Isolation of RNA Using Spin Technology

The protocol employed for RNA isolation uses column chromatography combined with spin technology (Figure 5.4). Once the samples have been properly disrupted and homogenized the chemical properties of the resulting samples becomes the same. Such isolation protocol of RNA can be used for all sample types included in this study: animal cells, plant cells, and animal tissues. The step-by-step Isolation of RNA using spin technology is described in the manual protocol available in Chapter 2 and comprises of steps 5 through 19.

Depending on the desired yield, elution steps can be repeated by using a second volume of RNase-free water or by retaining the previous flow-through. If a second fresh volume of RNase-free water is used, the yield becomes greater but the purified product has a lower concentration. If the flow-through is used for the second elution, the yield becomes lower but the product has a higher concentration. The volume vs. concentration tradeoff is dictated by the specific downstream protocols specifications and by cost considerations.

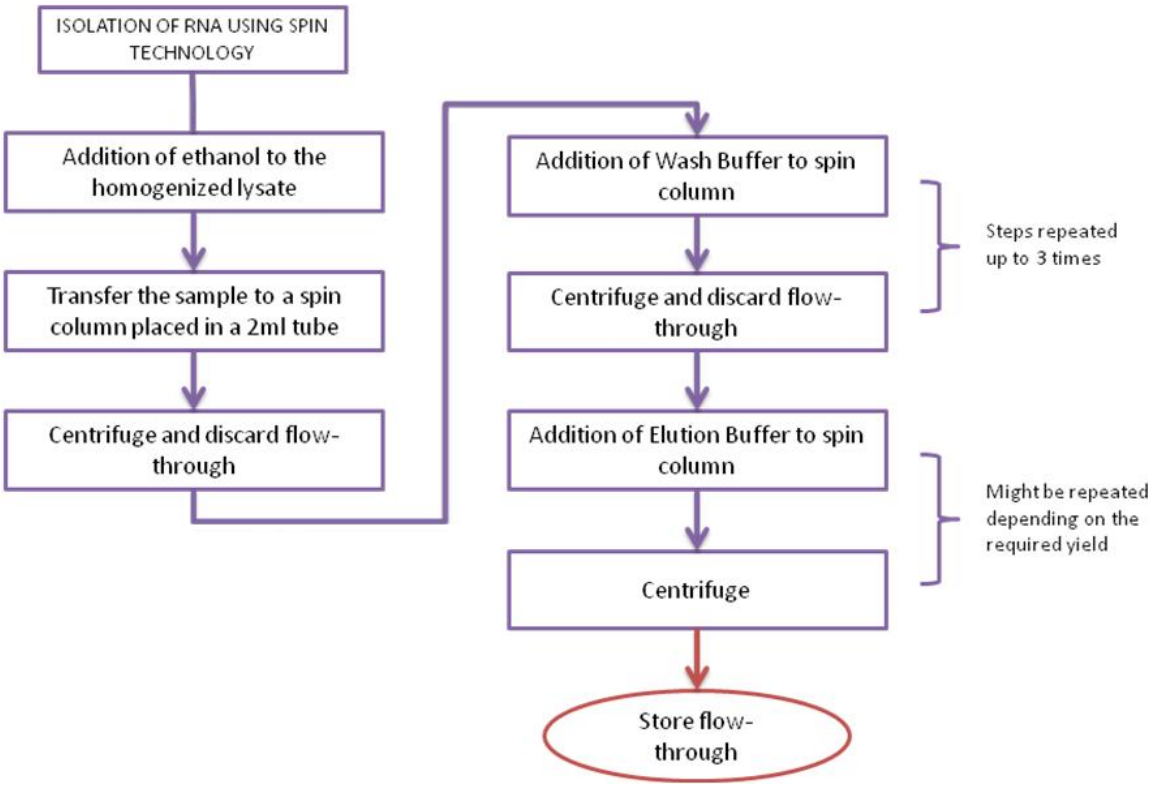

Figure 5.4 Isolation of RNA: spin technology

5.1.2. Purification of Total RNA from Plant Cells

The purification of total RNA from plant cells protocol is practically the same as the protocol used for animal cells with a few variations. This is so as it is assumed that the plant cells samples have been previously properly harvested, stabilized, and stored, and are ready for RNA purification without any other pre-processing actions (QIAGEN, 2001b). As already explained the isolation method described is that of column chromatography combined with spin technology (Figure 5.4). The differences with respect to the animal cells protocol reside in the disruption and homogenization stages.

Even though the disruption method for plant cells is basically the same as for animal cells: that of addition of lysis buffer and of vortexing (Figure 5.2), a different lysis buffer is used in each case. The purpose of the lysis buffer is to emulate the internal conditions of the cells in order to preserve the structure and functions of the cells content once these are lysed (Hayworth, 2014). As each type of cell has different internal conditions, different species require different buffer formulations.

One structural difference between plant cells and animal cells is that plant cells have cell walls and animal cells do not. This difference has a direct influence on the formulation of the lysis buffer and also on the homogenization procedure. As plant cell walls are in many cases made of cellulose, these are tougher than cell membranes that are present in both animal and plant cells. This might cause post-homogenization solid residues (debris) that may be present in the sample.

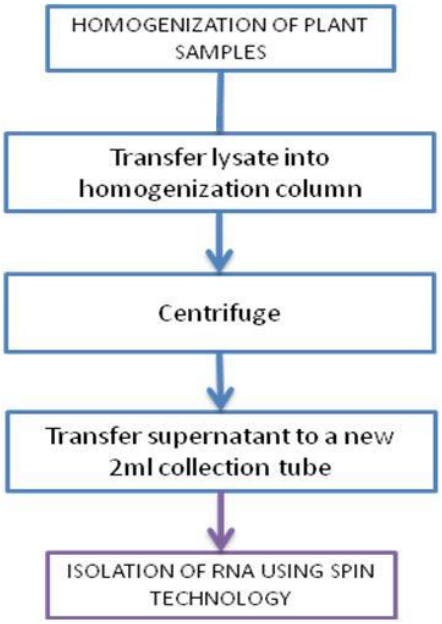

Figure 5.5 Homogenization method for plant cells: homogenization column and centrifugation

Homogenization columns are used as well to homogenize plant cells samples. Due to the relative toughness of the cell wall, an extra step is often added to the procedure that has already been discussed for animal cells (Figure 5.5). Once the lysate has passed through the homogenization column, the supernatant in the flow-through is transferred to a new 2ml collection tube and this supernatant is used for further RNA Isolation. As the supernatant is the liquid lying

above a solid residue following a centrifugation action (Merriam-Webster, 2014), the transferring of the supernatant to a new collection tube ensures that undesirable and useless solid particles are not present in the sample.

5.1.3. Purification of Total RNA from Animal Tissue

Purification of RNA from animal tissue is done by the same isolation procedure as those for animal and plant cells (Figure 5.4) with different disruption and homogenization method however. In tissues, cells are interconnected by a variety of structures, mostly involving extracellular matrix, proteins, and other molecules forming connecting bridges (University of Cambridge, 2012) (Kimball, 2014). In order To Isolate RNA, these structures and connections need to be broken down to get to the cells where the RNA resides. For this reason a more vigorous disruption method, typically mechanical disruption, needs to be employed to lysate and homogenize the sample.

In mechanical disruption the sample is lysed by introducing a stainless steel bead to the samples container and by shaking it at a relatively higher frequency (Figure 5.6). It is assumed that the sample has been previously properly harvested, stabilized, and stored, and it is ready for RNA purification without any other pre-processing actions (QIAGEN, 2001b).

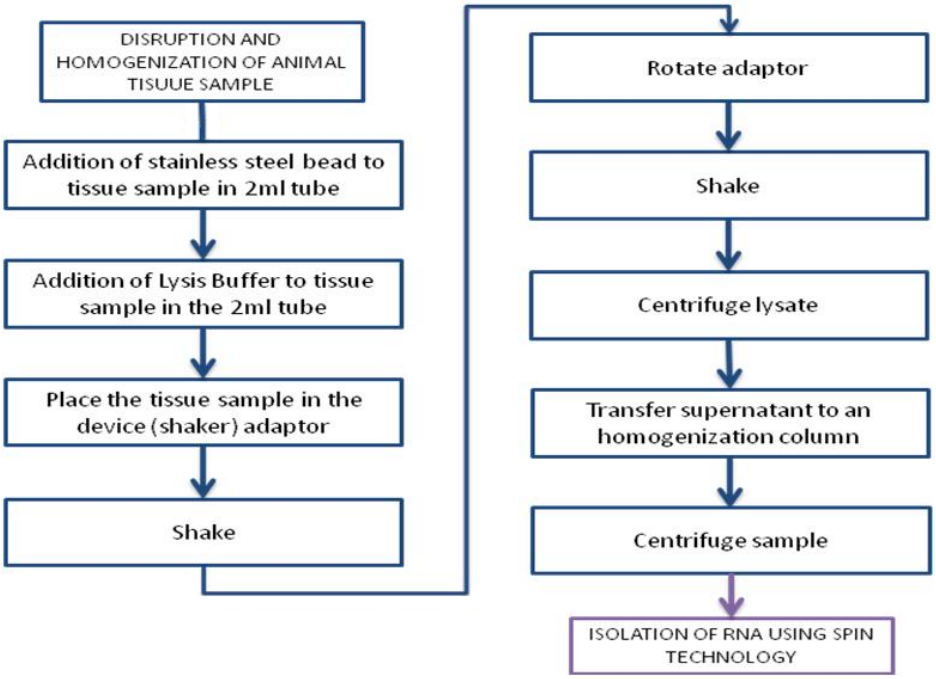

Figure 5.6 Disruption and homogenization of animal tissue

First, a stainless steel bead and the lysis buffer are added to a 2ml tube containing the tissue sample. The collection tube containing the tissue sample is placed into an appropriate shaker adaptor and the sample is then shaken at high frequency. As a result the tissue breaks down: the structures and connection between cells break, to allow disruption of the cell membranes. Next, the shaker adaptor is rotated and shaken again to ensure complete disruption of the tissue sample. Following the high frequency shaking, the collection tube is placed into a centrifuge to separate the solid phase from the liquid phase. The supernatant is transferred to a homogenization column and finally spun to get the desired homogenized sample. The 2ml collection tube is then used for RNA isolation.

5.1.4. Purification of Total RNA Summary

Table 5.1 Protocols used for Total RNA Purification for different starting materials

<table>
<tr><th></th><th>Animal Cells</th><th>Plant Cells</th><th>Animal Tissue</th></tr>
<tr><td rowspan="2">Disruption of initial sample</td><td colspan="2">Addition of lysis buffer to cells in 2ml tubes Note: Different lysis buffers are used for different cells</td><td>Addition of stainless steel bead to tissue sample in 2ml tube</td></tr>
<tr><td colspan="2">Vortex until cells are lysed</td><td>Addition of lysis buffer to tissue sample in the 2ml tube</td></tr>
<tr><td rowspan="7">Homogenization of sample</td><td rowspan="3">Transfer lysate into homogenization column</td><td rowspan="2">Transfer lysate into homogenization column</td><td>Place the tissue sample in the device (shaker) adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td rowspan="3">Centrifuge</td><td>Rotate adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td rowspan="3">Centrifuge</td><td>Centrifuge lysate</td></tr>
<tr><td rowspan="2">Transfer supernatant to a new 2ml collection tube</td><td>Transfer supernatant to an homogenization column</td></tr>
<tr><td>Centrifuge sample</td></tr>
<tr><td rowspan="10">Isolation of RNA using spin technology</td><td colspan="3">Addition of ethanol to the homogenized lysate</td></tr>
<tr><td colspan="3">Transfer the sample to a spin column placed in a 2ml tube</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge</td></tr>
</table>

Table 5.1 shows a summary of the steps used for the protocols for purification of RNA protocols from animal cells, plant cells, and animal tissue described in this section. The difference among these three protocols lies in the disruption of the initial sample and the homogenization of the sample.

5.2. Purification of genomic DNA

Genomic DNA, like total RNA, resides in the cells inside the nucleus and in the mitochondria. Therefore in order to get access to the DNA inside each cell, the cells have to be lysed and the sample has to be homogenized. Therefore, the purification protocol of genomic DNA has an almost identical three processes structure as the purification of total RNA. The main and obvious difference is that the third process is isolation of DNA instead of isolation of RNA. The three processes involved in purification of genomic DNA are: disruption of initial sample, homogenization of the sample, and isolation of DNA using spin technology (Figure 5.1).

The following sections focus on the differences between the purification of genomic DNA and purification of total RNA protocols for the different types of raw samples. The complete protocols are found in Appendix A.

5.2.1. Purification of Genomic DNA from Animal Cells

As the raw animal cells sample used for purification of genomic DNA from animal cells is the same as that for purification of total RNA, the cells are broken down using the disruption method previously described in Section 5.1.1: adding lysis buffer and vortexing the sample (Figure 5.2). As the internal conditions of the initial raw sample are essentially the same, the same lysis buffer is used for equal initial samples.

For homogenization that is done after the cells are broken down, a homogenization column is used as well to initially collect the available genomic DNA (Figure 5.3). Even though the homogenization method used for purification of genomic DNA is the same as for the purification of total RNA, the homogenization column used has different binding properties. For the DNA purification, as DNA is hydrophobic, the homogenization column used acts as a hydrophobic interaction chromatography column allowing genomic DNA to bind to it. As DNA binds to the

membrane it then can be washed, eluted and therefore isolated from the column. This is in contrast to RNA purification where a silica-base membrane is used.

The third process for purification of genomic DNA from animal cells is the isolation of DNA residing in the homogenization column by using Spin Technology. Once the sample has been disrupted and homogenized DNA is bound to the chromatography column, isolating DNA out of the column follows the same procedure for all types of samples: animal cells, plant cells, and animal tissue (Figure 5.7).

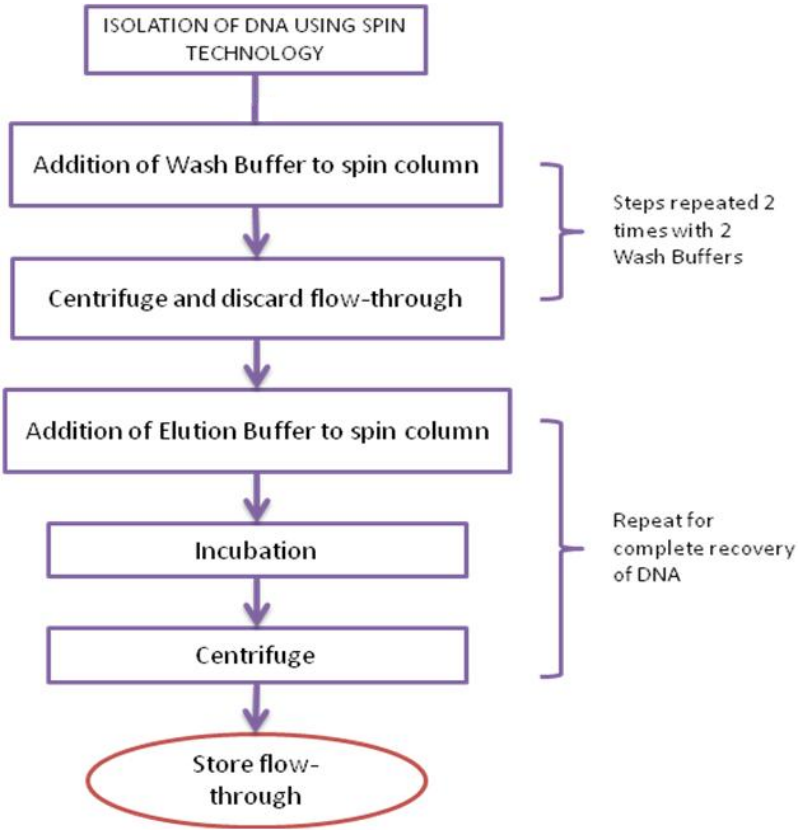

Figure 5.7 Isolation of Genomic DNA: spin technology

At this point in the process, genomic DNA is bound to the homogenization column, as do some other undesirable particles and therefore the first few steps are dedicated to washing unwanted particles out of the chromatography column to ensure that only DNA molecules are present. In order to do this, an initial wash buffer is added to the column to weaken the bond between undesirable particles and the column. This wash buffer is helped through the column by centrifuging of the homogenization column with a collection tube attached to it. The follow-through

is discarded and using a second wash buffer. Two more runs are performed (addition of wash buffer and centrifugation) to make sure that all the undesirable particles are washed of the membrane leaving only genomic DNA bound to it before starting any elution steps. This second run of washing steps takes longer and is very important since the purpose is to dry the membrane from any residual trances of salt or ethanol that may interfere with future downstream reactions. The elution steps are then carried out. First, elution buffer is added to the sample, the column is sealed and attached to a new collection/elution tubes. Next, the sample is incubated for a given period of time at a given temperature. Finally, the sample is centrifuged for a given period of time at a given temperature to elute the DNA out of the column and into the collection tube. For complete recovery of DNA these three steps are repeated a second time.

The repetition of the last three steps is carried out to ensure complete recovery of DNA but at a lower concentration since it would have the complete DNA material but in twice the volume of elution buffer. In the chemical and biological environment solutions can be dissolved but the reverse process is not possible. If the sample requires a higher concentration, complete recovery of DNA is not possible. A decision has to be made between higher concentration and higher yield, both are not possible.

5.2.2. Purification of Genomic DNA from Plant Cells

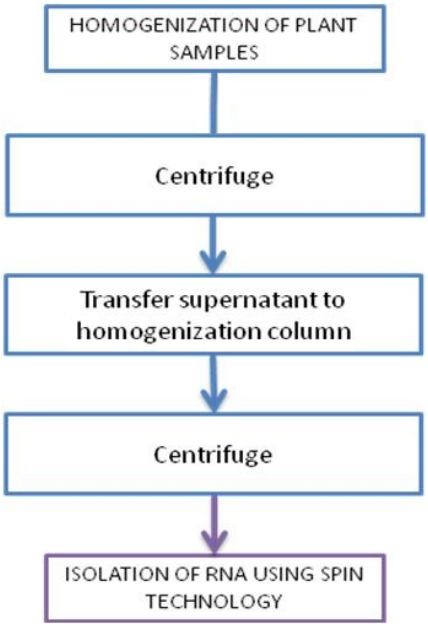

Figure 5.8 Homogenization method for plant cells

In the same way as with animal cells, as the initial sample is plant cells, the method previously described for disruption of plant cells, which is the similar to the one for animal cells is implemented keeping in mind to use the right lysis buffer (Figure 5.2).

For the homogenization method the lysate is centrifuged for a given period of time before the sample is transferred into the homogenization column. Thereafter, the supernatant is transferred into the homogenization column and a 2ml collection tube is attached to the column. Afterwards, the sample is centrifuged for a given period of time at a given temperature and the flow-through is discarded and the homogenization column is used for further DNA isolation (Figure 5.8).

The Isolation method for DNA in this case is the one previously described for isolation of genomic DNA using spin technology (Figure 5.7).

5.2.3. Purification of Gnomic DNA from Animal Tissue

The purification of genomic DNA from animal tissue can be viewed as a two processes protocol. First, as with purification of total RNA, the sample is disrupted and homogenized. As the initial samples used in both cases are the same, disruption and homogenization for purification of genomic DNA from animal tissue is performed as described in Section 5.1.3 for purification of total RNA from animal tissue (Figure 5.6). After disruption and homogenization is carried out, the DNA molecules are bound to the homogenization column and from there on the procedure previously described in Section 5.2.1 is followed for isolation of DNA using spin technologies (Figure 5.7).

5.2.4. Purification of Genomic DNA Summary

Table 5.2 shows the protocol for Genomic DNA Purification from animal cells, plant cells, and animal tissue. The differences between the purification of Genomic DNA from animal cells and plant cells are mainly three: a different lysis buffer is used even thought the process is the same; and different homogenization columns are used implying the inclusion of an extra step for plant cells. For animal tissue a different disruption and homogenization method is used based on mechanical disruption.

Table 5.2 Protocols used for Genomic DNA Purification for different starting materials

<table>
<tr><th></th><th>Animal Cells</th><th>Plant Cells</th><th>Animal Tissue</th></tr>
<tr><td rowspan="2">Disruption of initial sample</td><td colspan="2">Addition of lysis buffer to cells in 2ml tubes</td><td>Addition of stainless steel bead to tissue sample in 2ml tube</td></tr>
<tr><td colspan="2">Vortex until cells are lysed</td><td>Addition of lysis buffer to tissue sample in the 2ml tube</td></tr>
<tr><td rowspan="7">Homogenization of sample</td><td rowspan="3">Transfer lysate into homogenization column</td><td rowspan="2">Centrifuge</td><td>Place the tissue sample in the device (shaker) adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td rowspan="3">Transfer supernatant into homogenization column</td><td>Rotate adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td rowspan="4">Centrifuge</td><td>Centrifuge lysate</td></tr>
<tr><td rowspan="2">Centrifuge</td><td>Transfer supernatant to an homogenization column</td></tr>
<tr><td>Centrifuge sample</td></tr>
<tr><td rowspan="9">Isolation of DNA using spin technology</td><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Incubation</td></tr>
<tr><td colspan="3">Centrifuge</td></tr>
<tr><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Incubation</td></tr>
<tr><td colspan="3">Centrifuge</td></tr>
</table>

5.3. Overall Summary (Final Remarks)

Looking at the different protocols for Purification of Total RNA and Purification of Genomic DNA two main issues should be highlighted. First, the disruption and homogenization of a same kind of sample is the same. For example, the disruption and homogenization for plant cells if the same for both protocols: purification of total RNA and purification of genomic DNA. This also applies for animal cells and animal tissue. After the sample has been homogenized, what distinguishes if the sample is processed for RNA Isolation or for DNA Isolation is the utilization of the molecules bound to the column (DNA) or of the flow-through in the collecting tube (RNA). After homogenization, RNA flows through the homogenization column, and genomic DNA binds to the column. For RNA isolation the supernatant is used and for DNA isolation the column is used. Second, once the samples have been properly disrupted and homogenized, the steps for isolation itself are the same for every kind of initial sample.

The protocols described in this chapter, cover a large amount of different kinds of samples. For animal cells, as long as the cells have been properly cultured and grown, the protocol would be able to handle a large variety of samples. For plant cells, likewise if the samples have properly harvested, stabilized and stored, the protocols will handle the samples, keeping in mind that different cells may have different internal condition and thus will require different formulation of lysis buffers. In the case of animal tissue, the protocol is only able to handle fatty tissue and fibrous tissue as long as it has been previously harvested, stabilized, and stored. Process timings and lysis buffer varies for different samples.

It is also important to note that animal cells and plant cells could be disrupted using mechanical disruption. The method is not used because there are less time consuming, expensive, and appropriate methods for this kind of samples like the ones used in this chapter: use of lysis buffer and homogenization columns.

6. LABORATORY UNIT OPERATIONS (LUO)

6.1. Introduction

Every biotechnology protocol consists of multiple steps that have to be strictly followed to obtain a desired chemical or biological outcome. Twelve generic operations considered the basis for all protocols, have been identified in the literature for biotechnology. These are called Laboratory Unit Operations (LUOs) (Najmabadi, 2006a).

Automating a biotechnology protocol, as in automation done in any other industry, consists of automating the fundamental operations (LUOs) and integrating them into a production line (or cell). The equipment and technologies available for biotechnology can be classified into three groups:

Group 1 consists of all the manual equipment available for biotechnology, such as pipettes, syringes, and mortar and pestle.

Group 2 consists of the equipment that automates only one LUO at a time, such as a dispenser, centrifuge, robotic arm, and conveyors.

Group 3 consists of state of the art equipment that already automates and integrates more than one LUO, but is currently considered very expensive or customized equipment.

Since the focus of this dissertation, as stated in previous chapters, is automation for small-to-mid-size laboratories; this study contemplates only commercial off-the-shelf equipment that performs single operation (Group 2) assuming that such are available to small-to-mid-size laboratories. Later in this chapter, the discussion will be extended to include customized equipment and equipment that is available on the market that is capable of carrying out more than one operation (Group 3).

This chapter examines automation design issues related to the ways by which the basic twelve LUOs are employed to implement biotechnology protocols. First, it presents the LUOs classification found in the literature and the laboratory equipment needed for each operation. It

also describes the possible challenges encountered in the process of integrating the equipment into an automated solution. Second, it discusses the integration aspects and possible scenarios for automation implementation of equipment, and shows an implementation approach for representative cases. Third, the discussion is extended to include available customized solutions and more complex (and thus more expensive) instruments that are capable of performing more than one operation. Last, operations skipped in the earlier discussion of this chapter, are explained. These are considered important for a full description of the protocols in terms of LUOs.

6.2. The 12 Laboratory Unit Operations (LUO)

The concept of Unit Operation originates from the field of Chemical Engineering where such are considered to be basic steps in any process. A subset of all Unit Operations constitutes the Lab Operations Units for biotechnology (or in short LUO) as coined by (Najmabadi, 2006a). A set consisting of twelve LUOs for biotechnology was defined by modifying the original classification reported by the Zymark Corporation to consider the specifications and requirements of modern biotechnology laboratories (Zymark, 1988). The original table found in the Laboratory Robotics Handbook by Zymark Corporation was based on specifications for chemistry laboratories. This section describes the twelve LUOs as applicable to automation design issues in biotechnology protocols.

The twelve LUOs described by Najmabadi are the following: manipulation (LUO 1), liquid handling (LUO 2), separation or purification (LUO 3), conditioning (LUO 4), washing-drying (LUO 5), agitation (LUO 6), homogenization (LUO 7), breaking-fragmentation (LUO 8), weighing (LUO 9), measurement and direct detection (LUO 10), analysis and data extraction (LUO 11), and documentation (LUO 12). These LUOs have been numbered for easier identification throughout the chapter. It is important to keep in mind that not all twelve LUOs are necessarily used in every experiment and protocol; most of the protocols only use a small subset of the LUOs.

The vital issues to examine whenever automating any given LUO are the actual operation that takes place and the way it is done. The complete automated solution for any given LUO needs to take into account that for the operation to occur, the sample needs to be loaded into the equipment or transferred into the labware in which the operation takes place. Likewise, for the

sample to be ready for the next operation, it needs to be unloaded from the equipment or transferred to another labware. For example, whenever separation (LUO 3) is performed by means of a centrifuge, the automated solution needs to involve the loading and unloading (manipulation) of the labware (LUO 1) in and out of the centrifuge. This example demonstrates that by mere inclusion of a centrifuge (LUO 3) into the production line, the automation task remains incomplete. Such a separation solution needs to include a centrifuge (LUO 3) that is integrated with manipulation operations (LUO 1) for it to be a valid automated solution for sample separation. This logic can be extended to all LUOs. Each of these represents different challenges and different sets of LUOs that need to be integrated in order to find truly functional solutions. The following sections explore all possible combinations among the different LUOs and present an automated solution for these sets of LUOs.

6.2.1. LUO 1: Manipulation

Manipulation in biotechnology is the physical handling and transportation of labware, such as tubes, plates, caps, and arrays. This is better understood by dividing the manipulation operations into two categories: Category 1 includes the transportation of tubes, plates, and arrays among distinct locations (such as stationary equipment and stackers) and it is often realized by means of pick-and-place operations. Category 2 includes operations such as capping and uncapping of labware, lid openers and dispensing of labware. Manipulation in Category 2 also refers to manipulation and deposition of distinct or groups of molecules. The operations included are, for example, cloning, colony picking and array deposition. Operations included in Category 2 are carried out by standalone off-the-shelf equipment built for a given operation. This is stand alone equipment that needs to be loaded to execute the given operation and it can only perform the operation it was built for. Manipulation done at a molecular level is included in Category 2 of LUO 1 because the equipment that carries out such operations has the same characteristics and integration/automation issues as the rest of the equipment pertaining to this category.

For Category 1, in a manual experiment, the operator moves the labware to and from different pieces of equipment and loads and unloads the labware into and from the equipment. Whenever extrapolating a manual manipulation protocol step to an automated solution, there are,

as discussed in Chapter 3, two main solutions, and depending on the applications, one or both may be present in a production line: laboratory robotic arms (with cylindrical, articulated, or Cartesian geometries) or conveyers.

The equipment used for category 2 operations are lid openers, cappers/decappers, microarreyers, and colony pickers among others. Nowadays, laboratories tend to rely on automated equipment to perform these operations. Depending on the size of the laboratory, this includes equipment that only performs one operation where a human operator needs to load the labware. It also includes more complex equipment that integrates more than one operation and that may or may not have feeding mechanisms.

Assuming automation solutions for small-to-mid-size laboratories, this study only includes commercial equipment that can perform single type of operation assisted with basic or even nonexistent feeder mechanisms. The discussion of more complex manipulators support equipment is left as a future research direction.

There are two distinct ways to integrate the LUOs to the manipulation system. One is by means of conveyors operating jointly with feeder mechanisms. The second one is by means of pick-and-place devices or robotic manipulators. The appropriate manipulation systems used for different cases are explained in the following sections.

The manipulation operation describes how samples are transported throughout the processing line. The integration between different instruments in a process is determined by the operating principle of each of the equipment. Those requirements and constraints determine the need for pick-and-place devices, robotic transporters, and/or conveyors (Cardona, 2009). This will be further explained and discussed in Section 6.3.

6.2.2. LUO 2: Liquid Handling

The handling of all liquids, such as assay reagents, reaction reagents, buffers, samples, and the solutions, involves operations such as pipetting, dispensing and aspiration. The liquid handling operations would be involved in the adding of specific given volumes at a very high precision (low tolerances), the transferring of the reaction solution from one labware (usually a tube, a capillary tube, or an assay array) into another; and also in the mixing of some reactions if

performed by systematically pipetting in and out of the solution until the reaction solution is at the desired mixing quality. As with addition and measuring of volumes, operations such as transferring or mixing have to be performed with extreme care so as not to lose any part of the solution in the process. At such small volume levels, every tiny drop or particle can make a difference. Other than the level of precision needed for these operations, the automation itself of this LUO is straightforward: pipette tips or syringes are attached to pick and place devices that actually move solutions from one position to another (in the case of transferring, dispensing, aspiration, or addition). A concern that is deferred to future research is that of avoiding cross contamination between liquids.

An off-the-shelf dispenser that may or may not have feeder mechanisms is usually the automated equipment of choice used for liquid handling. The labware needed for the operation is placed underneath the actual dispenser's heads by means of feeder mechanism or pick-and-place devices. The dispenser may have movable heads equipped with pipette tips that perform dispensing and/or aspiration operations. The pipette tips are usually arranged in singles or in groups of 9 or 96 tips for conventional throughput operations, or in groups of 384 or 1536 for high throughput operations.

Whether or not the dispensing equipment has a feeder mechanism, dictates how it could be integrated into the production line. If the device is equipped with a feeder mechanism it can be easily integrated by means of conveyors, however if a feeder mechanism is not available, a pick-and-place device is needed.

6.2.3. LUO 3: Separation and Purification

Both Separation and purification involve the breakup of specific molecules from a solution. Separation is whenever the operation objectives are the molecules that are separated from the solution. It is called Purification whenever the operation objective is the fluid that becomes cleared of the separated molecules (Wikipedia, 2014).

There are many separation/purification methods such as filtration, centrifugation, magnetic separation, extraction, distillation, electrophoresis, and recrystallization. Each of these

methods is based on distinct physical or chemical principles and proper special purpose instrumentation.

These methods can be divided into three main groups. The methods used in the first group take place inside the labware containing the sample without having to add any extra reagent or element to the labware in order for the reaction to occur. This is the case of centrifugation or magnetic separation where the labware containing the sample is loaded into the equipment in order for the operation to take place. Methods in the second group involve the adding of elements into the sample's labware for the operation to take place. Recrystallization and magnetic beads are good examples for this group. For recrystallization, reagents and solvents are added in order for the operation and the reaction to take place. Magnetic beads have to be added before the actual operation is carried out and for the operation to be capable of happening. The third group includes methods, such as filtration, distillation, or electrophoresis, where the sample has to be transferred and has to flow through a device in order for the operation to take place. Afterwards, collection of the sample is executed depending on the desired product and the steps that follow the operation. The collection process too may take more than one step.

These methods can eventually be broken down into different steps, and there might be a need to use more than one piece of equipment to carry out the complete operation. Some of the steps can be recognized as other basic operations. Each device used for the operation has different requirements, which dictate if it has to be integrated by means of a pick-and-place device or a conveyer mechanism.

6.2.4. LUO 4: Conditioning

As its name clearly states, these operations modify and control the samples environment in order for a specific given reaction to occur. This is a very critical set of operations since, for example, enzymes need a specific temperature range for the reaction to occur. If the ideal environmental conditions are not met, the reaction will not take place.

There are two types of conditioning that must be carried out: the first is the actual conditioning of the laboratory (room); and the second is a conditioning that may or may not be required for a specific step or steps in the protocol.

Biotechnology laboratories need to be enclosed within clean rooms; there is a constant monitoring of the room temperature, pressure, humidity and particulate count. In this sense, laboratories are equipped with temperature, pressure, humidity and particle counting sensors. This first type of conditioning includes operations such as temperature control and atmospheric control. The integration of this type of conditioning LUO to an automated system is straightforward. The sensors are integrated and transfer the data directly to the laboratory management system. Since typically such environmental conditions have to be constant throughout the production line, there is no need to include or integrate this type of conditioning LUO to each step or another LUO for that matter.

The second type of conditioning occurs whenever specific steps of the protocol require special conditions. For example a given temperature needed for an enzyme to carry out a reaction or a centrifuge step that needs to be performed at a given temperature. For this second type of conditioning there are two possible scenarios for LUO integration. One occurs whenever a sample needs to be subjected to a different environment condition such as a temperature change for the enzymes to become active. In this case the LUO that needs to be integrated is the conditioning equipment by itself. Equipment used for this operations are hotplate heaters, incubators, refrigerators, ovens, freezers, vacuums fume hoods, and PCR machines. The second integration format for this type of conditioning operation is taken whenever another operation needs to be carried out at a different temperature, such as a centrifugation at a different temperature. In this case the integration of the conditioning LUO has to be thought out as integrated to the other LUO that actually does the operation. For example: a centrifuge that is placed inside a freezer or an incubator. Nowadays, some equipment such as centrifuges may have internal temperature control, and therefore the integration of such equipment is just as the integration of any other centrifuge.

6.2.5. LUO 5: Washing/Drying

An important operation is that of washing and drying of reusable labware such as tubes, plates and caps, pipette tips, syringes, etc. Washing and drying is performed by means of operations such as washing, sonication and vacuum drying.

This LUO is not part of any biotechnology protocol per se, but the operations have to be carried out if using reusable labware. In such cases the laboratory needs to be prepared to handle the washing and drying operations keeping in mind that these are time consuming operations, and that sufficient amount of labware needs to be available for performing the actual protocol. Even though washing and drying operations are necessary to prepare the labware for the process to be carried out, these operations do not need to be executed in real time as the protocol is carried out. These operations can be designed as a separate production cell that processes the labware by batches instead of one at a time. Therefore, it is not necessarily integrated into the production line for a given protocol. In any case, the loading and unloading of the labware to the equipment and the manipulation of the labware within this small production cell needs to be considered.

The washing/Drying LUO is special because it can be integrated as a separate cell on its own since it is not integrated into a protocol but into a production line; and it will be treated as such in the next section of this chapter.

6.2.6. LOU 6: Agitation

Agitation refers to the mixing of reagents together or with other molecules or particles. Agitation occurs whenever a reagent, molecule or particle is added to the solution at some step of a protocol and thereafter the new solution is mixed for further processing. Depending on the substance that is added to the solution and the type of upstream protocol agitation can be performed with various operations such as mixing, vortexing, shaking, and stirring.

For off-the-shelf available agitation equipment the operations require a pick-and-place device in order to load and unload the labware containing the samples into and from the equipment. Hence, a pick-and-pace device is needed to integrate the agitation equipment into the production line. Most if not all these operations, mixing, vortexing, shaking, and stirring, can be

performed within the agitation labware thus their integration does not require a liquid handling operation.

Equipment used to perform the various agitation operations includes vortex, shaker, automated rocker, and automated stirrer.

6.2.7. LUO 7: Homogenization

Homogenization in biotechnology refers to the making of a solution homogeneous by means of reducing the sample particles size thus creating a more uniform solution. Such particle size reduction is often done by cutting. Like previous LUOs and especially LUO 3 (separation and purification), homogenization can be carried out by numerous operations, and these operations depend on the type of sample in-hand and the protocol that is being performed. This operation can be carried out by means of sonication, grinding, cutting samples from gels, and homogenization columns.

As with LOU 3, these homogenization methods can be broken down into different steps depending on the characteristics of the operations. For example: the integration of an operation that takes place inside the labware that contains the sample, such as sonication, is different than the integration of an operation that needs to be transferred to a different labware or equipment such as a homogenizer column or a grinder.

Again, the complete solution for the integration of the LUO in any case needs to include the manipulation of the labware and the liquid handling of the sample and any other reagents needed to perform the operation.

6.2.8. LUO 8: Breaking/Fragmentation

This operation refers to the breaking of cells and to the dividing of large molecules. The latter is referred to as fragmentation. Methods used to perform this operation are among others lysis, mechanical or chemical cell disruption or fragmentation, and sonication.

Since some of the methods used for breaking and fragmentation are also used for homogenization, the same integration challenges listed for LUO 7 appear again for LUO 8. Even though LUO 7 and LUO 8 at a first glance appear to be the same kind of operation, as their goals are different, there is a need to distinguish between these LUOs. LUO 7 of homogenization aims

to have a homogeneous solution by reducing the larger particles sizes; whereas LUO 8 of breaking and fragmentation only intends to break or fragment the sample. For example, lysis breaks the cell membranes allowing access to the RNA and DNA inside the cells.

Other operations used for this LUO present the same integration challenges such as loading and unloading the equipment, and the handling of reagent needed for chemical disruption, for instance.

6.2.9. LUO 9: Weighing

Weighing is the measurement of samples mass. This operation requires specialized equipment since the masses that are considered in biotechnology are very small (in the order of magnitude of as low as hundreds of picograms). This operation is performed by direct measurement.

The equipment is pre-calibrated using the same empty labware used later on in the protocol. The requirement that needs to be considered in this case is the interface of the equipment with the production line. There are some off-the-shelf scales available with feeder mechanisms that can be integrated by means of pick and place devices, but there are others that do not include a feeder mechanism making a pick-and-place device the only solution to load and unload the sample to and from the equipment.

6.2.10. LUO 10: Measurement and Direct Detection

Measurement and direct detection includes the direct measurement of physical properties such as pH, conductivity, absorbance and fluorescence. Some techniques used to measure physical properties in biotechnology protocols include the adding of special dyes to the reaction. Thereafter, by measuring the fluorescence and/or absorbance of the reaction and comparing it to a known desired outcome a physical property can be quantitatively measured. Other operations encompassed in direct measurements and detection are: pH measurements and conductivity measurements. There are two purposes for measuring physical properties of a reaction or sample. The first one would be to ensure and control that the protocol is carried out correctly. The second one, present in downstream protocols, is whenever the goal of the protocol is to measure the results of different reactions like those carried out in screening for drug discovery.

Whenever the purpose of measuring is to ensure that the protocol is being carried out correctly, measurements such as fluorescence, absorbance, and pH measurement involve the addition of reagents or manipulation of the sample that is not actually part of the protocol. For this reason these measurements are only performed to a sample of the samples, and these samples are kept for the record or are discarded. Such samples are not allowed to proceed to the next automation line station since these have been tampered with. This is one of the reasons why in small-to-mid size biotechnology laboratories a close loop control local to each station is almost never performed: controlling the quality of the samples may mean the discarding of a relatively significant amount of product.

Whenever the measuring of the physical properties of the reaction is part of the protocol as in screening for drug discovery, the measurement is performed to all samples in order to process them. Downstream protocols are beyond the scope of this dissertation.

Integrating LUO 10 into the production line is significantly different from that of other LUOs. For once, after the operation takes place the sample is not put back into the production line, but instead it is stored in a different storage unit (or discarded). Also, it is not performed to all of the samples available. Even though closed loop control of the protocol is out of the scope of this dissertation, the challenges and requirements to integrate such operations are taken into consideration in the next section (Section 6.3).

6.2.11. LUO 11: Analysis and Data Extraction

Analysis and data extraction is defined as analysis of the outcome of the experiment using specific instrumentations. LUO 11 Operations may include mass spectrometry, biochip analysis, and sequencing, among others. This LUO is more commonly used in sample processing protocols (downstream protocols). Operations that perform analysis and data extraction are carried out by specialized equipment that performs each of the operations - mass spectrometer, biochip screener, and sequencer. Labware needs to be loaded and unloaded to and from the equipment by means of pick-and-place devices, or if the equipment has feeding mechanisms it can be integrated to a conveyor.

6.2.12. LUO 12: Documentation

Documentation in the context of biotechnology refers to the creating of appropriate data documentation for retrieval and further analysis and reporting. This operation is performed usually by database management software applications that can be commercially available solutions or customized for each laboratory. The integration of the documentation and data collection is available if management software is integrated to the equipment needed for the protocol. The level of completeness of the integration and the information collected is directly proportional to the integration of the management software to the equipment.

6.2.13. Summary

Table 6.1 summarizes the 12 LUOs essentials, with the different operations and methods that carry out each, the equipment used in the methods, and the challenges encountered whenever integrating these LUOs into a production line.

Table 6.1 Summary of LUO methods and integration features

LUO	Methods	Equipment	Integration Features and Challenges (*italic*)
LUO 1	Manipulation of labware	Pick-and-place devices, robotic manipulators and conveyors.	• The only standalone operation.
	Capping/decapping, manipulation and deposition of molecules	Lid openers, cappers/decappers, microarreyers, and colony pickers among others	• Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware to the equipment.
LUO 2	Dispensing reagents, pipetting and aspiration samples.	Dispensers, pipettors and syringes.	• Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware to the equipment.

LUO 3	Centrifugation and magnetic separation.	Centrifuge and magnetic stand.	• Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware to the equipment. • *Needs to be loaded by means of pick-and-place devices.*
	Recrystallization and magnetic beads.	Magnetic stand and different chemical compounds.	• Operations take place inside the labware containing the sample • A liquid dispensing operation is needed for the operation to take place. • The sample needs to be loaded into different equipment. • *These operations may consist of multiple steps and rounds*.
	Filtration, distillation, and electrophoresis.	Filtration systems.	• Samples need to be transferred to flow through the actual device where the operation takes place (e.g.: a filter). • It needs to be collected and transported to the next station.
LUO 4	Temperature control, atmosphere control, thermal cycling, and incubation.	Temperature, atmosphere, humidity sensor.	• The room has all necessary sensors integrated to the laboratory management software
		Hotplate heater, incubator, refrigerator, oven, freezer, vacuum, fume hood, and PCR Machine.	• Condition changes only for a given step of the protocol • The labware containing the sample needs to be manipulated inside the different equipment
LUO 5	Washing and drying.	Washer, water bath, vacuum, autoclave.	• Operations are not part of a biotechnology protocol. • *Integration of these operations is done as a cell separated from the production line.*
LUO 6	Mixing, vortexing, shaking, and stirring.	Vortex, shaker, stirrer, and rocker.	• Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware to the equipment. • *Needs to be loaded by means of pick-and-place devices.*

LUO 7	Homogenization, gridding, cutting samples from gels	Homogenization columns, blender, Mortar & pestle, chopper, razor blade,	• Samples need to be transferred to flow through the actual device where the operation takes place. • It needs to be collected and transported to the next station.
	Sonication	Sonicator	• Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware to the equipment.
LUO 8	Lysis and chemical cell disruption.	Vortex, shaker.	• Liquid handling operations are needed • Loading of the labware into the equipment by means of pick-and-place action. • It needs to be collected and transported to the next station.
	Mechanical cell disruption or fragmentation and sonication.	Mechanical disruptor, mortar & pestle, and sonicator	• Transfer operations of the sample are needed • Loading of the labware into the equipment by means of pick-and-place action. • It needs to be collected and transported to the next station.
LUO 9	Quantitative measurement of sample mass.	Balance and scale.	• Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware to the equipment.
LUO 10	Fluorescence and/or absorbance, pH measurements, and conductivity measurements.	UV lamp, UV cross linker, luminometer, fluorescent optical scanner, chemical reagents.	For upstream protocols: • Measurements are not part of the protocol • Different chemical reagents are added. • *Tested samples do not continue into the production line* • *Only a sample of the samples is tested.* For downstream protocols: • Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware to the equipment.

LUO 11	Mass spectrometry, biochip analysis, and sequencing, among others	Mass spectrometer, biochip screener, and sequencer, among others	• Operation carried out by different equipment • Integration solution needs to include the loading and unloading of the labware.
LUO 12	Documentation and data collection	Data base management software	• Software Management System

6.3. Classification of LUO Integration Architectures

So far the discussion has focused on how basic operations are carried out by equipment controlled manually by a human operator and how LUOs are the smallest units for such operations. For biotechnology automation a need to integrate the LUOs into a production line arises. Many of the LUOs simply cannot constitute standalone automation stations. For instance LUO3 (separation and purification) must be assisted by LUO1 (manipulation) and LUO2 (liquid handling) (Cardona, 2014a).

The most basic level of system integration involves the interfacing of LUOs one to another. LUOs such as LUO1 and LUO2 are common interfacing-type LUOs. In any biotechnology automation line each of the stations executes typically one of the LUOs described above, whereas one or more other LUOs may play supportive roles.

Since the operation of the equipment needed for implementing a specific LUO may require multiple extra steps, a new building block that appears to be suitable for the automation environment for biotechnology protocols is created named Automation Module (AM). An AM is a collection of LUOs that constitute a building block for automation design. An AM eventually could become one of the automation line stations.

Even though the twelve LUOs execute different lab operations, similarities of the extra steps required to perform the various operations can be established. In other words, the LUOs might use the same feeding mechanisms, carry out the same type of operation within the equipment, may include liquid handling operations to transfer samples among labware, and so on. In this section, these similarities are exploited and grouped together in order to find what Automation Modules (AM) represent and how these could incorporate the different LUOs into a production line.

Need to keep in mind two features of AMs: First, each AM does not map (or represent directly) each LUO. This means that LUO1 cannot become AM1, LUO2 cannot become AM2, and so forth. This is because the AM encompass not only an LUO but the steps required to carry it out. Second, as we try to find common functional characteristics of existing LUOs we realize that different LUOs may be implemented using the same AM description. This means that each LUO is to be represented by a single AM. How these common or similar characteristics are exploited to describe different AMs, and how these AMs are interconnected to one another is explained below.

6.3.1. Conceptualization of Automation Modules (AMs)

In order to understand the relevance of the AM concept it is essential to fully appreciate the automation role of LUOs. LUOs can play three main roles within a protocol: Those that the main function of which is to move labware, compounds, reagents, and samples around the production line; those that process the samples to obtain required outcomes; and those that handle the data measurements and data collection of the protocols.

For integration and automation purposes, LUOs that process the samples or collect data can be addressed using the same approach. In these operations samples need to be loaded or transferred to a different equipment or location. Manipulation operations that require specific equipment, such as a capper/decapper, lid openers, microarrayers, or colony picker, can also be automated using the same approach.

Three LUOs LUO 1, LUO 5, and LUO 12 require special consideration because of the following reason:

Automation of LUO1 (manipulation), the main function of which is to move or transport labware throughout the production line, has long been solved by means of robotic manipulators, pick-and-place devices, or conveyors, and all their possible combinations. The implementation of any transportation method consists in integrating the right type of transporter with the correct end-effector to the production line.

For LUO 5, as washing and drying are operations that may be performed off-line, whenever it is carried out off-line it can be seen as a separate specialized production cell. This

specialized production cell does not need to be integrated into the sample production line. For this scenario a Special Case Automation Module was designed and will be discussed in Section 6.3.6.

The integration of the documentation LUO (LUO 12) depends on the management software used, if any, throughout the production line. Documentation depends solely on the software and communication between different instruments, and it does not involve physical operations performed to the samples. For these reasons there is no definition of an AM for this LUO.

The process of formulating AMs starts by realizing that LUOs tend to use more than one method to carry out the various operations (For example, there are many separation methods.). As a result the similarities of the extra steps required to carry out the various LUOs depend on the specific equipment needed to perform one of the methods of a given LUO. Different methods may have constraints and requirements that are common and can be grouped together to formulate a single automated solution - Automation Module - for each group of methods. Taking into consideration the 12 LUOs and the typical recurrent methods found in the literature, seven AMs are created. Table 6.1 helps in identifying common features and requirements among the methods used for implementing the 12 different LUOs. Additional AMs may be defined as needed.

Each AM is a collection of sequenced LUOs that can be integrated into an automated production line to perform one of the different methods that constitute a given LUO. This collection of LUOs and their appropriate sequence is formulated following three steps:

1. Breaking down the method that is used into a set of steps required to fully perform the needed operation
2. Identification of the LUO needed to implement each step
3. Formulation of the set of all LUOs needed to create an AM and their order.

The following example illustrates the procedure (Figure 6.1):

Taking as an example the task of filtration (which is one of the methods that can carry out LUO3), the steps that have to be followed to perform separation are:

1. A sample arrives at the station.
2. The sample is removed from the original labware and put into a container containing the filter.
3. The sample advances through the filter and as it comes out it is collected in a new labware that is placed underneath the filter.
4. The new labware containing the filtered sample is taken to the next automation station or to an interim location (just to assure that the sample has left the station).

As shown in the example above, the main separation action only takes place in step 3, but the sample has to go through two prior steps. These prior steps can too be expressed in term of some of the LUOs. That is, the sample arrives at the station by means of labware manipulation (LUO 1), and is thereafter transferred to a second container by means of liquid handling (LUO 2). At the end, following the filtration step, the sample has to leave the station by means of labware manipulation (LUO 1). In summary, to automate a separation operation (LUO 3) using filtration method, the Automation Module consists of the following set of operations: LUO 1, LUO 2, LUO 3, and LUO 1 (Figure 6.1).

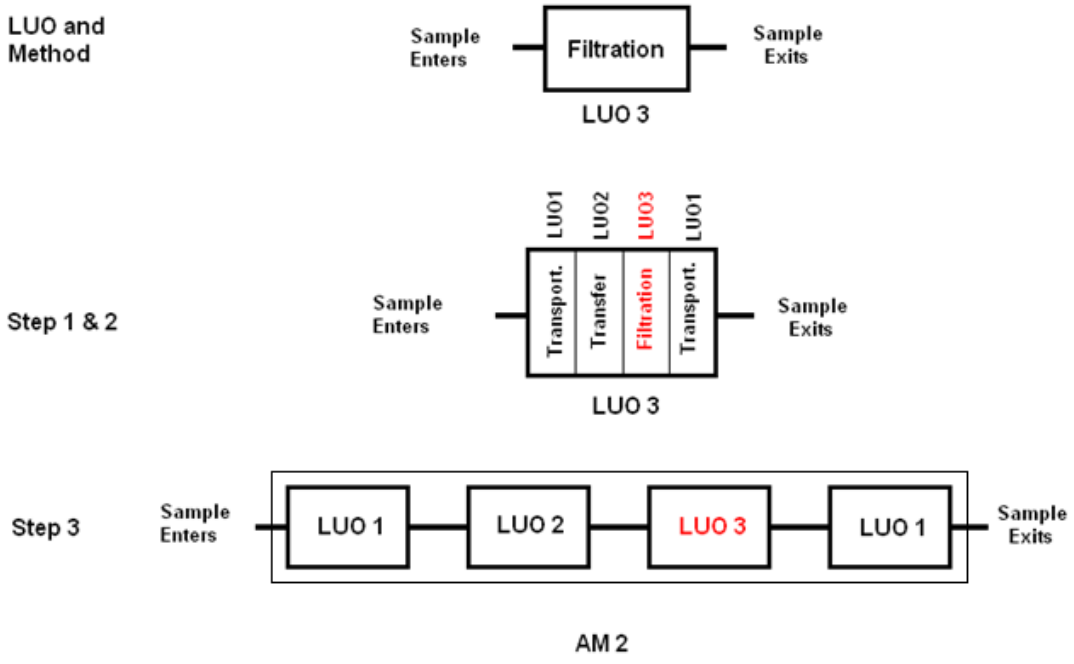

Figure 6.1 Conceptual diagrams for the formulation of AM 2 for filtration

After a LUO is broken down into fundamental steps and a plausible Automation Module is defined, the equipment needed for implementing each of the steps is identified.

Going back to the filtration example, the specific requirements of the actual filtering operation dictate how the labware manipulation and liquid handling operation are carried out.

Every Automation Module may be implemented in many different ways. In the case of the filtration example, the Automation Module may consist of a manipulator, a dispenser, a vacuum filter, and may or may not include an additional pick-and-place device. Depending on the particular dispenser that may be selected, there may be additional pieces of equipment that may be added to the AM, such as a conveyor.

Following the definition of the entire automation line by means of a sequence of Automation Modules and following the selection of specific pieces of equipment needed to implement each AM, the integrating of consecutive AMs one to another by means of LUO1 (manipulation) needs to be addressed for each pair of consecutive AMs. For each pair there are two possible scenarios. The first is whenever the AMs implementation equipment modules share the same manipulation equipment.

The second scenario is whenever the pair of consecutive AMs have different manipulation hardware. In such a case a sample exchange interface sub-station needs to be included. For example, a pick-and-place device may drop the labware on a conveyor belt, or pick up the labware from a conveyor to load the station of interest.

Following the aforementioned approach, LUOs can be classified into basic Automation Modules. Table 6.1 shows that LUOs 1, 2, 6, 9, and 11 operations are carried out each by one method and have one main implementation feature. Therefore a single Automation Module may be used to implement all these operations. LUOs 1, 2, 6, 9, and 11 operations all take place inside specific equipment and loading and unloading of the labware needs to be taken into consideration for the automation architecture. LUOs 3, 7, and 8 have implementation methods that carry out these operations with the same structure as LUOs 1, 2, 6, 9 and 11 and therefore will use the same AM. These three LUOs (3, 7, and 8) also have more than one solution for other methods used to carry out the operations. Finally, Table 6.1 shows that LUOs 4, 5, and 10, may

have special cases implementations. In summary, four general Automation Modules and three special cases Automation Modules are drawn to integrate any of the twelve biotechnology LUOs.

This section discusses the Automation Modules created by analyzing common equipment and methods for the different LUOs. Arena® is used to validate the feasibility of the implementations. Arena® is discrete event simulation software widely used in automation (Kelton, 2002).

6.3.2. Automation Module 1 (AM1)

The most common situation is whenever an LUO can be fully implemented by means of standalone equipment. As such AM1 is created just to provide the manipulation actions needed at the input and output of the respective stations. That is, the sample is placed into the equipment prior to the processing and is to be removed following the processing. The general structure of AM1 is: LUO1 → LUOX → LUO1, where LUOX may be one of the following LUOs that are known to be implemented by standalone equipment {LUO1, LUO2, LUO3, LUO4, LUO6, LUO7, LUO8, LUO9, LUO10, LUO11}. Which form of LUO 1 is used depends on the requirements of the equipment itself that is used to implement LUOX and the stations that precede and follow that AM. The equipment is chosen among off-the-shelf equipment that is available to perform the needed task.

Some off-the-shelf equipment such as centrifuges and vortexes require top loading and others may require side loading. The latter may need no feeder or any other interface mechanisms with conveyors or may require a pick-and-place device. On the other hand, some equipment, such as dispensers or incubators, may use a feeder mechanism that allows an easy interaction and integration with a conveyor belt, such as dispensers or incubators.

In outlining all the valid LUOX choices we keep in mind common available equipment for the various LUOs that could be integrated into AM1 as shown below.

- LUO 1: capper/decapper, lid openers, and any other manipulation that requires an off-the-shelf equipment to handle labware.
- LOU 2: dispensers, pipetters, and syringes if the operation performed is an addition of a reagent or compound to the labware that already holds the sample.

- LOU 3: centrifugation and separation via magnetic stands.
- LOU 4: hotplate heaters, incubators, refrigerators, ovens, vacuums, and PCR machines (among others).
- LUO 6: shaker and vortex (among others).
- LUO 7: sonicator.
- LUO 8: sonicator and vortex.
- LUO 9: balance.
- LUO 10: luminometer, optical scanner among others (not for upstream protocols)
- LUO 11: mass spectrometer and biochip scanner.

AM1 has the following steps:

1. The labware arrives at the station and is placed or loaded into the equipment.
2. The equipment performs the operation.
3. The labware is removed or unloaded from the equipment and is moved to the next station.

This configuration can be evaluated using the Arena® model shown in Figure 6.2.

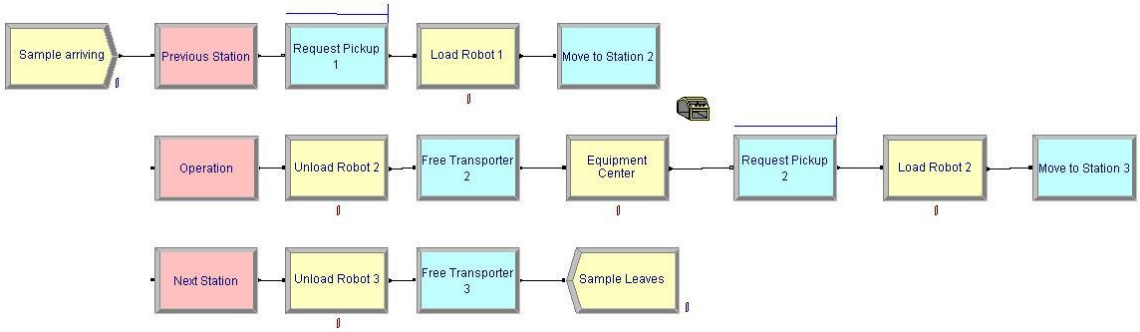

Figure 6.2 Arena® model for Automation Module 1

Figure 6.2 shows the Arena® module used to evaluate AM1. This module simulates a sample arriving from a previous station (LUO 1), the operation that takes place, and the sample leaving the station (LUO 1).

6.3.3. Automation Module 2 (AM2)

The second type of AM involves tasks in which the sample must be transferred from its original container into another one. Its structure is LUO1 → LUO2 → LUOX → LUO1, where

LUOX is one of the set of LUOs {LUO3, LUO7}. These are Filtration and Homogenization tasks, respectively. We recall that such tasks are not performed inside the labware containing the sample, but there is a need for the sample to be transfer into another device for the operation to take place.

AM2 involves the following steps:

1. The sample arrives at the station.
2. For filtration, the sample is taken out of the original labware and is put into the container where the filter is present; for homogenization the sample is dispensed into a homogenization column.
3. The sample passes through the filtration or homogenization device and thereafter it is collected in a new labware placed underneath the filter or homogenization column.
4. The new labware containing the filtered sample is taken to the next station or to the next interface location (sample leaves the station).

The Arena® model used to evaluate and validate this implementation is shown in Figure 6.3.

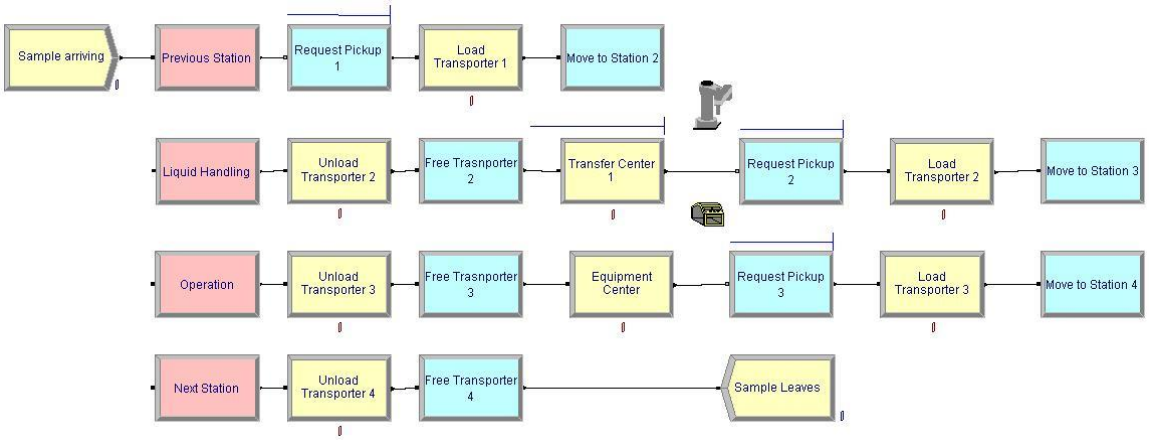

Figure 6.3 Arena® model for Automation Module 2

In this case, the actual operation takes place in step 3, but the manipulation of the labware and the transferring of the liquid sample are essential for the integration of the operation into a production line.

6.3.4. Automation Module 3 (AM3)

AM3 deals with operations that require the addition of reagents into the sample before the actual operation can take place. Therefore the general structure of AM 3 is LUO1 → LUO2 → LUO1 → LUOX → LUO1, where LUOX is chosen from the set {LUO3, LUO8}. There are magnetic bead separation and chemical disruption, respectively. The Arena® model used to evaluate this Automation Module is shown in Figure 6.4.

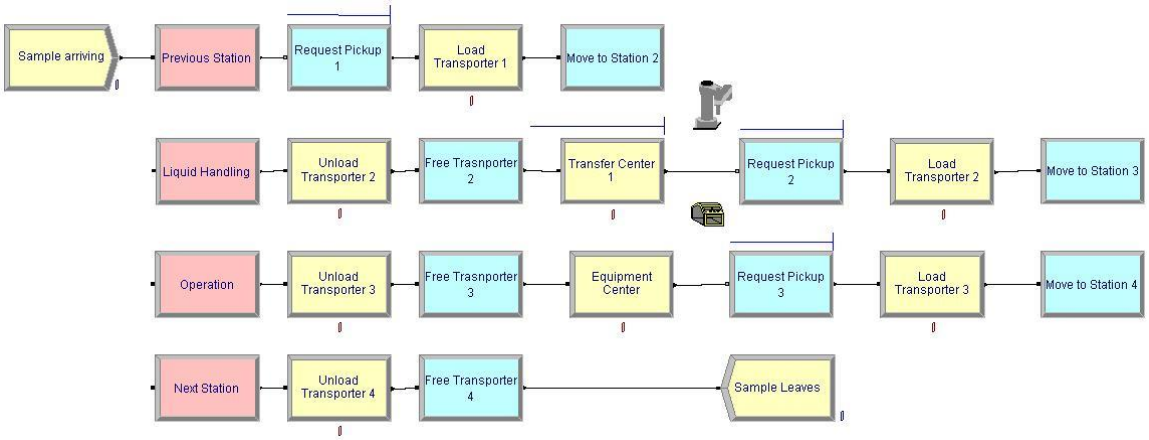

Figure 6.4 Arena® model for Automation Module 3

The steps of AM3 are:

1. The labware containing the sample arrives at the station and is placed within reach of the dispensing device.
2. In the case of magnetic separation, a bead is dispensed into the labware containing the sample; in the case of chemical disruption, the chemical disruptor is dispensed into the sample.
3. The labware with the sample is moved and placed into the equipment that actually performs the needed operation, a magnetic stand for LUO 3 or a mechanical disruptor for LUO 8.
4. The (disrupting or separating) equipment performs the operation
5. The labware is removed from the equipment and placed in a dispenser to be able to separate the liquid phase of the reaction.
6. The sample of interest (liquid phase) is transfer into a fresh tube (labware) to take it to the next station.
7. The labware is transported to the next station.

6.3.5. Automation Module 4 (AM4)

AM4, like AM3, involves the addition of a reagent prior to additional manipulation that moves the sample to the main operation site. However, the reagents must be removed from the sample labware prior to sending the sample to the next automation station.

Operations that are integrated using AM4 are:

- LUO 7: grinding, and cutting
- LUO 8: fragmentation

The general structure of AM 4 is therefore: LUO1 → LUO2 → LUO1 → LUOX → LUO1 → LUO2 → LUO1, where LUOX is taken from the set {LUO7, LUO8}.

Two different modules exist because even though step 2 for both modules is LUO 2, in module 3 the operation needed is the dispensing of a reagent into the labware containing the sample; and in this case, the operation that is needed is the transferring of the sample from the current labware, to where the operation actually takes place. Figure 6.5 shows the Arena® module that simulates and validates this solution.

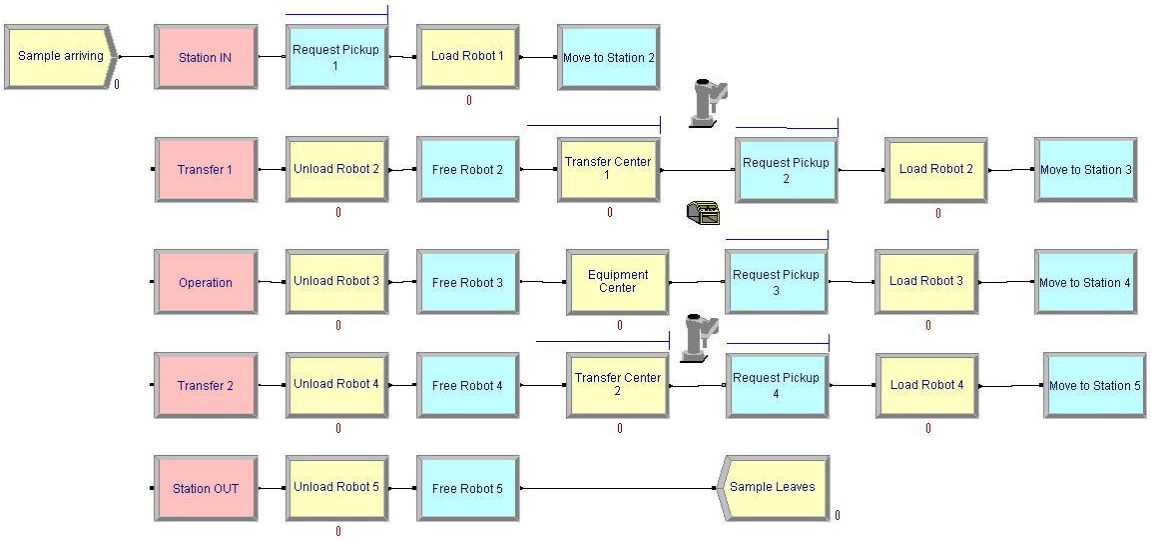

Figure 6.5 Arena® model for Automation Module 4

These operations follow the following common steps:

1. The labware arrives at the station.
2. The sample is transferred into a different labware that can attach to the equipment.

3. The labware with the sample is placed into the equipment that actually performs the operation.
4. The equipment performs the operation
5. The labware is removed from the equipment.
6. The sample of interest is transfer into a new tube (labware).
7. The labware is transported to the next station.

6.3.6. Automation Module 5 (AM5)

Automation Module 5 integrates LUO 5 (washing and drying). As explained earlier, washing and drying of reusable labware operations are not part of any biotechnology protocol. Such operations have to be performed in order to have the labware available for one of the actual protocols. The general structure of AM5 is: LUO1 → LUO5 → LUO1 → LUO5 → LUO1, where the first LUO5 involves a washing operation and the second LUO5 involves a drying operation.

The Arena® module used for this implementation is shown in Figure 6.6.

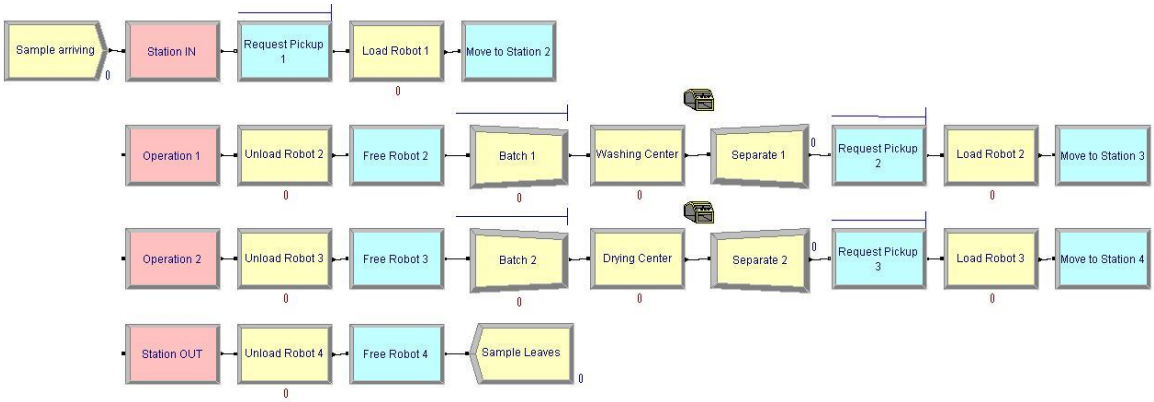

Figure 6.6 Arena® model for Automation Module 5

AM5 has the following step sequence:

1. The labware arrives at the station and is placed inside the washing equipment.
2. The equipment performs the washing operation.
3. The labware is taken out of the washing equipment and is placed into the dryer.
4. The equipment performs the drying operations.
5. The labware is removed from the equipment and taken to a storage unit.

There is some of-the-shelf equipment that can actually perform both operations (washing and drying) within the same machine. In such a case the functional structure of AM5 is: LUO1 → LUO5 → LUO5 → LUO1.

6.3.7. Automation Module 6 (AM6)

As explained earlier there are two types of conditioning referred to by LUO 4. The first is the actual environmental conditioning of the entire laboratory space, and the other involves the conditioning of specific steps in the protocol.

For the condition of the laboratory one adds heating units, air conditioning units, and humidifiers, among others. These are controlled by means of environmental sensors located around the lab such as thermometers, humidity sensors, pressure sensors, and so on. For this type of conditioning no Automation Module are needed.

The conditioning of a specific step of a protocol can be merely that the sample needs to be incubated, or heated, or frozen, or any other environmental condition change applicable just to the sample. This is carried out with AM1 as already explained because the sequence of steps for such operations is simply: LUO 1 → LUO 4 → LUO 1.

There is a second possibility wherein an operation needs to be carried out at a different set of environmental conditions. For instance, a centrifugation that needs to be carried out at lower than room-temperature conditions. In such a case, there are extra steps that may be needed to perform that same operation that includes the LUO of interest along with LUO 4. Extra manipulation steps may be required for this purpose. There are various examples for this kind of conditioning and therefore it may be hard to generalize the steps. For example: LUO1 → LUO4 → LUO1 → LUO3 → LUO1 → LUO4 → LUO1.

6.3.8. Automation Module 7 (AM7)

The last module is used to integrate LUO 10 (Measurement/Direct Detection) in upstream protocols. Most measurements and direct detections are carried out to monitor chemical reactions by means of optical methods. In order to obtain any kind of optical results, an activator or a detection reagent, foreign to the actual protocol needs to be added to the sample. Because of the detection reagent that is added to the sample, the sample is used to monitor the process to this

point, but it cannot continue in the automation process as it has been tainted. Therefore not all samples go through such monitoring process. Integration of these operations is considered a special case since monitoring for upstream protocols are not performed to all the samples in order to get a desired detection, and since once a sample is taken out of the production line it is not put back into normal circulation. These operations are mostly done for monitoring purposes in upstream protocols and can be represented by AM7. An example of such operations is pH measurement.

The sequence of LUOs for this module is the same as for AM1, but after the sample is processed it does not continue into the next station of the protocol, but instead it would go to a different storage unit or to a discarding station.

This operation follows the sequence:

1. One sample every given number of samples arrives at the station and is placed inside the equipment.
2. The equipment performs the measurement operation
3. The sample is removed from the equipment and is stored or discarded.

Figure 6.7 shows the Arena® model for validation of Automation Module 7. The sequence of LUOs for this operation is therefore LUO 1, LUO 10, and LUO 1.

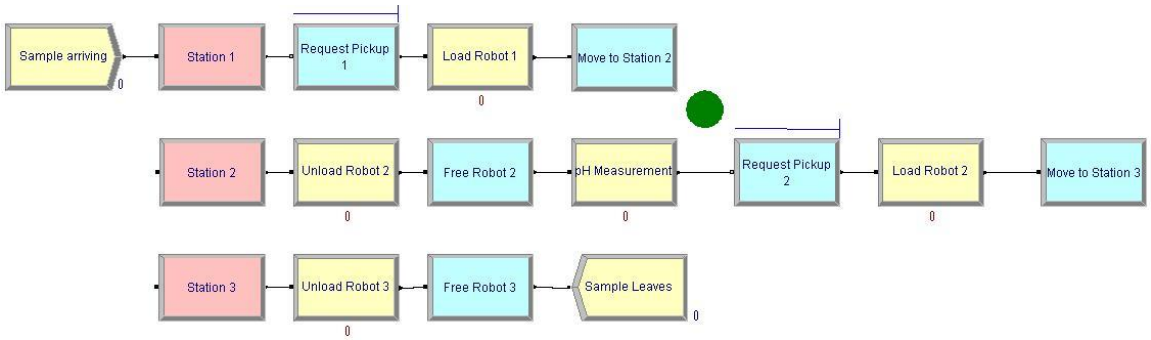

Figure 6.7 Arena® model for Automation Module 7

6.3.9. Summary of Automation Modules

Table 6.2 contains a summary of the Automation Modules defined in Section 6.3. It shows all LUOs and typical operation methods, and their corresponding Automation Module. It also describes the integration scheme used for each module.

Table 6.2 Summary of Automation Modules

Automation Module	LUOs and methods that can be implemented with the AM	Implementation Scheme
AM1	LUO1 – Manipulation that requires equipment to handle labware LUO2 – Liquid handling LUO3 – Centrifugation and separation via magnetic stands LUO4 – Conditioning that requires equipment (only conditioning) LUO6 – Shaker and vortex LUO7 – Sonicator LUO8 – Sonicator and vortex LUO9 – Balance LUO10 – Luminometer, optical scanner LUO11 – Mass spectrometer and biochip scanner	LUO1→LUOX→LUO1
AM2	LUO3 – Filtration LUO7 – Homogenization	LUO1→LUO2→LUOX→LUO1
AM3	LUO3 – Magnetic beads separation LUO8 – Chemical disruption	LUO1→LUO2→LUO1→LUOX→ LUO1
AM4	LUO 7 – Grinding and cutting LUO8 – Fragmentation	LUO1→LUO2→LUO1→LUOX→ LUO1→LUO2→LUO1
AM5	LUO5 – Washing and drying	LUO1→LUO5→LUO1→LUO5→ LUO1
AM6	Conditioning of a specific operation. The implement scheme depends on the operation that requires the special conditioning.	
AM7	LUO10 – pH measurements	LUO1→LUO10→LUO1

6.4. Integration of Automation Modules

A manual labor biotechnology production line consists of a sequence of LUOs needed to implement a complete protocol. Automated biotechnology production lines consist therefore of a sequence of automation modules indicated by each of the protocol LUOs. Automation Modules are the building blocks of a full automation line design. An automation line consists of AMs that are integrated one to another. Such integration involves equipment matching (specifically of LUO1 manipulation equipment), removal of redundant equipment and possibly equipment sharing.

To integrate these Automation Modules into a production line and be able to simulate a complete biotechnology protocol the following steps have to be followed:

1. Identifying the LUOs needed in the protocol
2. Identifying the method needed to implement each LUO.

3. Identifying the Automation Module that could be used to integrate the operation
4. Depending on the transportation and manipulation of the labware throughout the protocol, the integration of consecutive AMs is optimized.

Automation Modules help pointing out possible equipment sharing for the transportation and manipulation of the labware equipment. The selection of equipment for such transportation and manipulation tasks is a decision that the designer has to make depending on the transportation and manipulation requirements of consecutive stations and the overall automation line architecture that would be employed throughout the system. For example, if two consecutive stations have to be loaded/unloaded by means of pick-and-place devices due to the way the equipment operates, the designer would have at least two configuration selection options: The first is to use dual pick-and-place devices, one serving the manipulation needs of each instrument, and a conveyor for the transportation of the labware between stations. The second option is to use a robotic manipulator that carries out both operations: labware manipulation and transportation for the two stations. Each option involves different workspace for the manipulators, different footprint for the solution, different costs, and even different throughput.

In Chapter 8 a selection criterion is defined that involves cost, footprint and throughput to detect among the candidates solutions which one would have the best return on investment. The designer, after having to consider multiple possible candidate layouts for any sub-system of the production line derived from the implementation of the AMs should employ the selection criterion to quantitatively evaluate each candidate solution and choose the best one.

6.4.1. Complete Example: Preparation of RNA from Tissue Culture Cells

Preparation of RNA from tissue-culture-cells protocol that was chosen as an example to illustrate the usefulness of the Automation Modules concept consists of a 20 step procedure (Florida Atlantic University, 2006). One initially identifies the LUOs of the protocol, and there is no need to get into great detailed consideration of the specific components, reagents or quantities used (for more details of the protocol refer to Chapter 2). At this point in time, the protocol is only looked at from an automation viewpoint and the hardware and equipment needed to create such an automated solution.

The 20 steps for the preparation of RNA are as we recall:

1. Retrieval of one tube full of frozen cells.
2. Addition of 350 µl of RLT buffer followed by gentle vortexing until cells are lyzed.
3. Transferring of the lysate into a QIAshredder spin column placed in a 2ml collection tube.
4. Centrifuging of the tube for 2 minutes at 14,000 rpm
5. Addition of 350 µl of 70% ethanol to the lysate in the collection tube.
6. Transferring of the combined 700µl, including any precipitate that may have formed, to an RNeasy mini column in a 2ml collection tube.
7. Centrifuging of the tube for 15 seconds at 14,000 rpm.
8. Discarding of the flow-through and reattaching the collection tube to the mini column.
9. Addition of 700µl of buffer RW1
10. Centrifuging of the tube for 15 seconds at 14,000 rpm.
11. Discarding of the flow-through and the collection tube.
12. Transferring of the RNeasy mini column to a new 2ml collection tube.
13. Addition of 500µl of the wash buffer RPE onto the column.
14. Centrifuging of the tube for 15 seconds at 14,000 rpm.
15. Discarding of the follow-through and reattaching the collection tube to the mini column.
16. Addition of another 500µl of the wash buffer RPE onto the column.
17. Centrifuging of the tube for 2 minutes at 14,000.
18. Addition of 50µl of RNase free water and attaching a new collection tube.
19. Elution of the RNA from the column by centrifugation for 2 minutes at 14,000 rpm.
20. Storage of the collecting tube.

First, in order to identify each LUO, each step and operation needs to be understood. Thereafter, the method used for each operation needs to be identified. The next step is to identify

which Automation Module (AM) could be used for each of the LUOs and steps present in the protocol. The LUOs, methods and AM for each step are:

- Step 1: LUO 1 – pick-and-place device
- Step 2: LUO 8 – Vortex. Implementation: AM3
- Step 3 and Step 4: LUO 7 – Homogenization column. Implementation: AM2
- Step 5: LUO 2 – liquid dispenser. Implementation: AM1
- Step 6: LUO 2 – pippetor. Implementation: AM1
- Step 7: LUO 3 – centrifuge. Implementation: AM1
- Step 8: LUO 1 – labware dispenser. Implementation: AM1
- Step 9: LUO 2 – liquid dispenser. Implementation: AM1
- Step 10: LUO 3 – centrifuge. Implementation: AM1
- Step 11: LUO 1 – labware discarder. Implementation: AM1
- Step 12: LUO 1 – pick-and-place device
- Step 13: LUO 2 – liquid dispenser. Implementation: AM1
- Step 14: LUO 3 – centrifuge. Implementation: AM1
- Step 15: LUO 1 – labware dispense. Implementation: AM1
- Step 16: LUO 2 – liquid dispenser. Implementation: AM1
- Step 17: LUO 3 – centrifuge. Implementation: AM1
- Step 18: LUO 2 – liquid dispenser. Implementation: AM1
- Step 19: LUO 3 – centrifuge. Implementation: AM1
- Step 20: LUO 1 – pick-and-place device

Following the identification of the AMs, check for possible redundancy in transportation steps and equipment must commence. In order to remove redundancies each Automation Module is expanded to a sequence implementation steps. The manipulation of labware between consecutive AMs is examined and some of these steps are thereafter removed. For example, looking into steps 1 to 3 we obtain:

- Step 1:
    1. The labware is retrieved from a storage unit

- Step 2:
    2. Labware arrives to the station and is placed in the dispenser device.
    3. The chemical disruptor is dispensed into the sample.
    4. The labware with the sample is placed into the vortex.
    5. The equipment performs the operation
    6. The labware is removed from the equipment and placed in a dispenser.
    7. The sample of interest is transfer into a new labware.
    8. The labware is transported to the next station.
- Step 3:
    9. The sample arrives to the station.
    10. The sample if dispensed into a homogenization column.
    11. The sample goes through the homogenization column device and is collected in a new labware placed underneath the homogenization column.
    12. The new labware containing the sample is taken to the following station.

Inspecting the above example, the labware that is retrieved from the storage unit (step 1), can go straight to the dispenser device (step 2); that way only one manipulation operation is performed. Hence in order to remove redundancy only one pick-and-place device is needed between Station 1 and Station 2. Likewise, as the sample is transferred into a new labware and the labware goes to the next station in step 2, that same pick-and-place device deposits the sample into the dispenser unit, step 3. These three steps with 12 sub-steps can be reduced to 10 sub-steps:

1. The labware is retrieved from a storage unit and is placed in the dispenser device.
2. The chemical disruptor is dispensed into the sample.
3. The labware with the sample is placed into the vortex.
4. The equipment performs the operation
5. The labware is removed from the equipment and placed in a dispenser.

6. The sample of interest is transfer into a new labware.
7. The labware is transported to the next station.
8. The sample is transferred into a homogenization column.
9. The sample goes through the homogenization column device and is collected in a new labware placed underneath the homogenization column.
10. The new labware containing the sample is taken to the following station.

By doing so with the remaining 17 steps and stations, the redundancy can be further eliminated as some manipulation equipment can be shared among several stations that are in proximity in terms of the protocol step order.

A friendly CAD tool that could assist in the assessment of the various design configurations is Arena®. Whenever building Arena® models for the different Automation Modules the simulation process stays essentially the same: each module is placed one underneath the other in the correct order and the repeating blocks may be erased to remove redundant equipment.

The complete Arena® model for this case study is presented in Figure 6.8.

The simulation was carried out setting the simplest possible manipulation method for each of the instruments used. This comprises of conveyors for the operations that allow a use of conveyors, and of simple pick-and-place device for the instruments that absolutely demand manipulation by means of a robot equipment such as dispensers and automatic pipettes can be fed by means of conveyors and feeder mechanisms, whereas centrifuge and vortex need pick-and-place devices. This overall system candidate configuration includes one storage unit, six dispensers, one vortex, two automatic pipettes and column stands, and six centrifuges, a conveyor belt that goes around all 17 stations, and 7 pick-and-place devices that manipulate the labware from the feeders mechanisms at the conveyors to the vortex and centrifuges. Each pick-and-place device serves one station and one instrument.

The above model has been simulated and validated using Arena® and its feasibility has been established. This Preparation of RNA model is capable of processing 234 samples per 8 hours shift, taking on the average 124 seconds per sample.

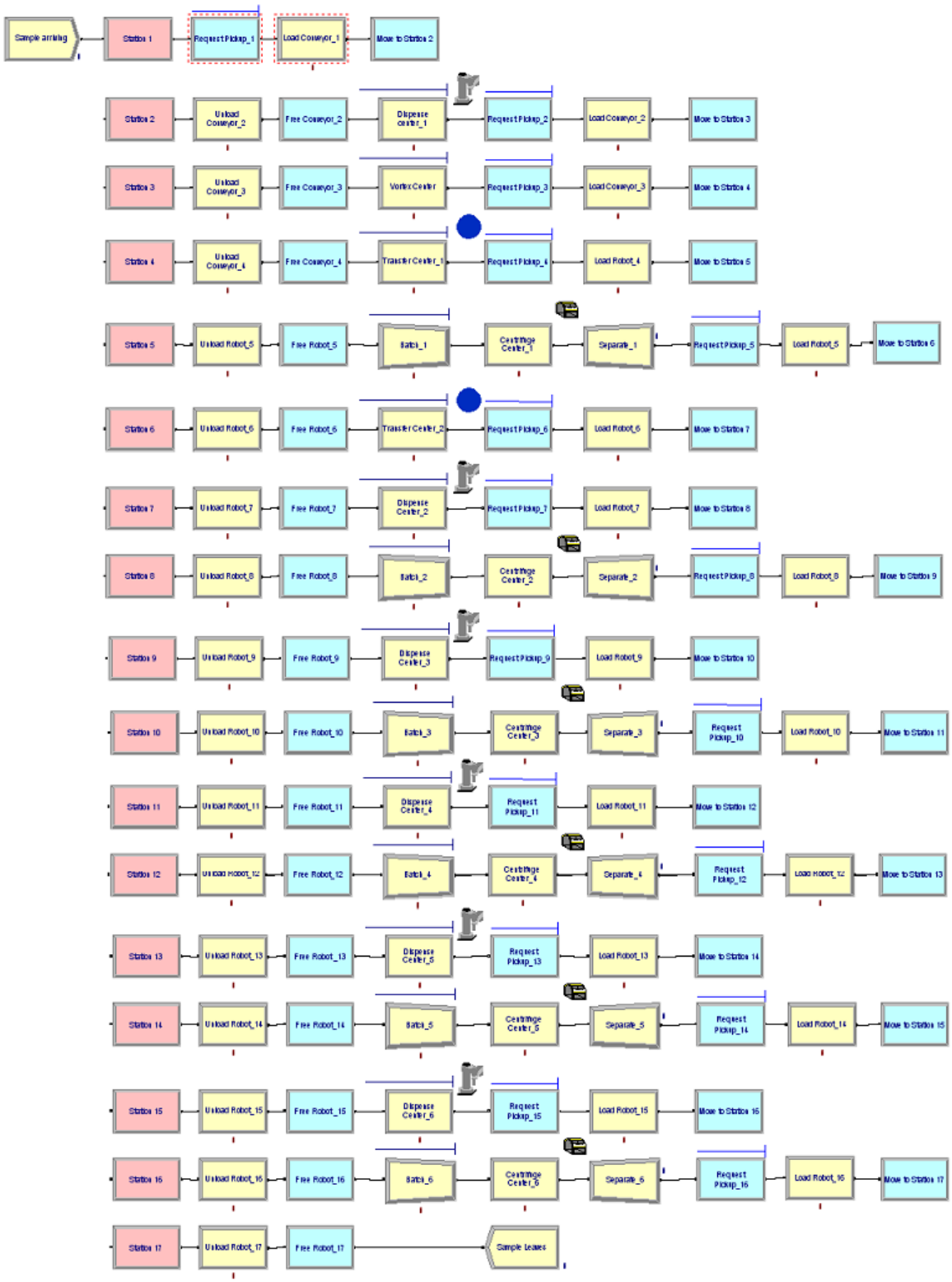

Figure 6.8 Arena® model for Preparation of RNA

6.5. Integration of Available Customized Solutions and Multi-LUO Equipment

In Section 6.1, the equipment and technologies available for biotechnology were classified into three groups. Group 1 that is not covered in this dissertation as it involves manual instruments. The off-the-shelf automation equipment in Group 2 was thoroughly covered throughout Sections 6.2 to 6.4. This section discusses automation issues that relate to Group 3 which consists of state of the art equipment that already automates and integrates more than one LUO, but is considered costly or highly customized.

Some systems described in the literature are so complex and involved that entire protocols can actually be run on them as table-top solutions, not to mention entire systems solutions like the HTS-TRAC™ Tecan system already describe in Section 3.3.3. There are other instruments that automate more than one operation but not complete protocols. These two kinds of equipment are discussed in the following subsections (Tecan, 2008).

6.5.1. Complete Table-Top Solutions

There exist systems that carry out specific protocols or a family of protocols. As these systems are already serve as complete solutions no or very little integration is further needed. These automation solutions are comparably significantly more expensive than designing a home-made solution for a given protocol.

6.5.1.1. DNA Purification System – Thermo Scientific™

The DNA Purification System by Thermo Scientific is a table-top system capable of running an entire protocol. The DNA Purification System is an automated tool for DNA purification protocols in which beads based methods use magnetic particles. It is advertized as being able to automate all steps involved in the purification of DNA samples: the binding of DNA to magnetic beads, the washing of the bound material, elution from the solid phase, and transfer to the final destination plate (Thermo Scientific, 2014). The vendor argues that the system is adaptable to any magnetic bead-based separation method (Figure 6.9).

This system consists of Standard lab table (2.00 × 2.44m) ; F3 robot system (6 axes of motion) ; Servo Gripper with Microplate Fingers; Magnetic Bead Settling Stations (four) ; 120 Plate Carousel; Microplate Feeding Stations (120-plate capacity); Dispenser with bead

dispensing module; Plate Delidder with barcode reader; Microplate Shaker with pneumatic lock for reliable plate placement and retrieval (four stations); Biotek EL 405 Plate Washer (96-well with bead settling plate); Robbins Hydra-96™ Pipettor with bead settling plate; Polara™ software and instrument interfaces (Thermo Scientific, 2014). All the components of the system are developed and built by Thermo Scientific.

Figure 6.9 DNA Purification System – Thermo Scientific™ (Thermo Scientific, 2014)

The DNA Purification System, equipment-wise, costs much more than a system made of Group 2 equipment. The integration of Group 3 systems could be seen as “plug-and-play” whenever combined done with other equipment obtained from the same vendor. However it can get tedious whenever trying to add pieces of equipment purchased from different companies.

This system is advertised to be a high throughput system as it can process up to 14 96-well plates per hour. It can work unattended for 24 hours. That means than in a 24 hour period of time it would be capable of processing 336 plates. This is not entirely true as the carousel can

only hold 120 plates. That would mean that it could only run for about 8 hours or at least that every 8 hours an operator has to restock the carousel with fresh plates. In Section 6.4.1 for the example of RNA purification from cultured cells, the system was able to process 234 samples every 8 hours.

6.5.2. Integration of Multi-LUO Equipment

There has been rapid development in recent years in laboratory automation equipment and systems. There are now quiet a few systems by Beckman Coulter, Tecan, and Thermo Scientific, among others, that are capable of automating more than one biotechnology lab operation. Such turnkey systems tend to be costlier than systems designed and put together using off-the-shelf equipment. For example, a state of the art luminometer that integrates two or three dispensers can cost much more than a traditional luminometer combined with traditional dispenser.

For integration purposes, such complex automation system may be visualized as a “black box” (Figure 6.10) even though the designer needs to understand the operational details of the equipment included in the system.

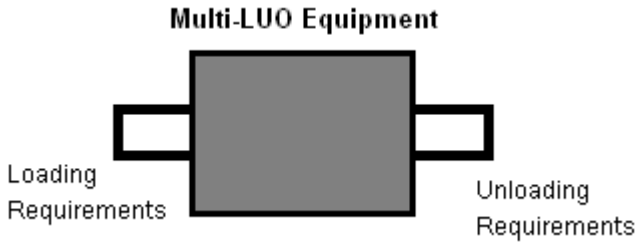

Figure 6.10 Multi-LUO equipment define as black boxes

Taking for example the Centro LB 960 Microplate Luminometer by Berthold Technologies which has three built in reagent injectors (Berthold Technologies, 2014); if the designer decide to adopt the equipment into the automated production line, the only concern is about the loading and unloading of samples to and from the equipment. At this point the designer need not be concerned with the internal operation of the instrument. This example will be elaborated more in a later subsection.

As the equipment for integration is conceived as a black box, such black box can be viewed as any other specific LUO and therefore it can be integrated via AM1 : LUO1 → Multi-

LUO Equipment → LUO1. The type of method used for transportation depends on the loading and unloading requirements of the equipment. The process of integrating such special AM1 to a production line is slightly different than the one followed for the Automation Modules described in Section 6.4.

This time following the identifying of the LUOs needed in the protocol, the designer needs also to identify the LUOs executed by the available multi-LUO equipment and the sequence by which the operations are carried out. This sequence has to follow the same sequence or a subset of the same sequence in the protocol to be able to use the equipment. Biotechnology protocols have sequential steps with a sequence that cannot be altered. If the sequence of steps executed by the multi-LUO equipment matches the sequence of steps needed by the protocol the complex unit is adopted and the automation integration proceeds as outlined earlier. For better understanding of the process, Section 6.5.2.1 illustrates it with an example.

Once the multi-LUO equipment is integrated into the production line, the management software of the automation system takes over. It is recommended to use equipment taken all from the same vendor throughout the production line as it facilitates integration.

6.5.2.1. Integration Example: Centro LB 960 Microplate Luminometer

The Centro LB 960 Microplates Luminometer by Berthold Technologies (Figure 6.11) is a universal microplate Luminometer that has three built-in reagent injectors. These injectors can be set up to dispense multiple activators or detection reagents and variable volumes running from 10 to 100µl. In addition, the equipment includes a waste pump which safely discards any extra unwanted fluids; and a temperature controlled microplate compartment to ensure stable conditions whenever sensitive environmental conditions are needed for avoiding degradation of certain specific enzymes and cells. The equipment can create controlled temperatures up to 42 ^{o}C. It has a small footprint, 390 x 450 x 350 mm (w x d x h), including all its important features, and it was designed with a philosophy of allowing maximum use of available bench space. It works with a sample format of 96-well plates as well as 384-well plates. As the equipment was designed keeping a modular concept in mind, it can be easily integrated, both in terms of software and hardware, into existing robotic and liquid handling systems.

Most of such next generation laboratory equipment has been developed with two main properties: the use of universal standard formats and a modular systems concept for easier integration.

Figure 6.11 Centro LB 960 Microplate Luminometer (Berthold Technologies, 2014)

In order to illustrate how the integration procedure for multi-LUO equipment is done the following hypothetical generic protocol is described:

1. Retrieval of one microplate with a cell X sample
2. Addition of Y µl of reagent A to the microplate.
3. Incubation for Z minutes.
4. Addition of Y µl of detection reagent B to the microplate.
5. Testing for fluorescence absorbance
6. Storage of microplate.

The LUOs for these the above six steps and their respective Automation Modules are identified as follows:

- Step 1: LUO1 – pick-and-place device
- Step 2: LUO2 – liquid dispenser. Implementation: AM1
- Step 3: LUO4 – incubator. Implementation: AM1
- Step 4: LUO2 – liquid dispenser. Implementation AM1
- Step 5: LUO10 – luminometer. Implementation AM1
- Step 6: LUO1 – pick-and-place device

A conceptual block diagram of the theoretical protocol is shown in Figure 6.12.

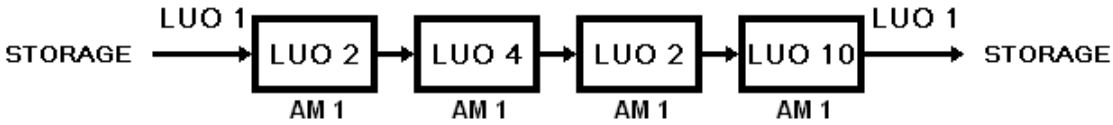

Figure 6.12 Conceptual block diagram for the theoretical example

The next step of integrating the Center LB 960 Microplate Luminometer is to identify the LUOs this equipment can execute. As the Center LB 960 Microplates Luminometer has three built-in reagent injectors and a luminometer, this instrument would be able to carry out liquid handling operations and direct detection operations, in other words, LUO2 and LUO10. The LB 960 can execute numerous different sequences involving up to three liquid handling operations and three direct detection operations depending on the protocol requirements and how it is program. Some examples of sequences that it could perform are:

- LUO2→LUO10→LUO2→LUO10
- LUO2→LUO2→LUO10
- LUO2→LUO10
- LUO2→LUO10→LUO2→LUO2→LUO10, and so on.

Looking at the theoretical protocol, it has one of these sequences performed by the multi-LUO equipment listed above which is the sequence: steps 4 and 5 (LUO2→LUO10). The sequence of operations that the LB 960 can be programmed to carry out and that is present in the theoretical protocol would be LUO2, LUO10 (Figure 6.13)

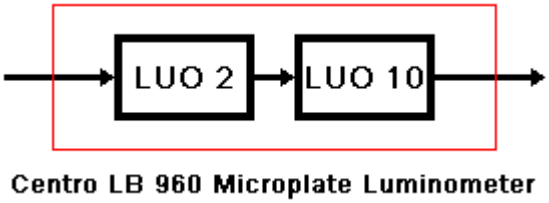

Figure 6.13 Conceptual block diagram for the Centro LB 960 Microplate Luminomenter for the sequence that will be replace in the theoretical protocol

Once the sequence to be replaced is identified (LUO2, LUO10), the LUO representing the sequence are replaced by a black box representing the new instrument that would take their place. This black box would be integrated into the production line with AM1. The new conceptual

diagram for the theoretical protocol after replacing the sequence of interest is shown in Figure 6.14.

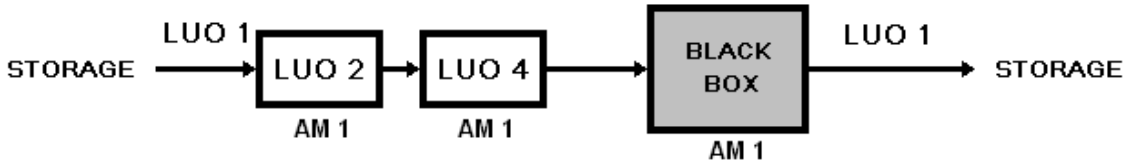

Figure 6.14 Conceptual block diagram for the theoretical protocol after replacing the equipment

The LUOs and Automation Modules for the theoretical module are identified now as follows:

- LUO1 – pick-and-place device
- LUO2 – liquid dispenser. Implementation: AM1
- LUO4 – incubator. Implementation: AM1
- Black Box (LB 960) . Implementation: AM1
- LUO1 – pick-and-place device

After removing redundancies, steps 3 to 6 of the theoretical protocol would be executed following the sub-steps in Table 6.3:

Table 6.3 Sub-steps for steps 3 to 6 for the theoretical protocol when using only equipment from Group 2 (left), and equipment from Group 2 and 3 (right)

Steps 3 to 6 for Theoretical Protocol Using Equipment from Group 2	Steps 3 to 6 for theoretical Protocol Using equipment from Group 2 and 3 (LB 960)
1. The labware is transported form Incubator and placed in a liquid dispensing unit.	1. The labware is transported form Incubator and placed into the LB 960
2. The reagent is added to the sample.	2. The Equipment performs the operation.
3. The labware is transported from the liquid dispenser unit and is placed in the luminometer	3. The labware is transported from the luminometer to the storage unit.
4. The Equipment performs the operation.	
5. The labware is transported from the luminometer to the storage unit.	

The complete conceptual solution found for the Theoretical Protocol when using the LB 960 (Figure 6.14) was model using the Automation Modules described in Section 6.3. The final model used to simulate these six steps is shown in Figure 6.15.

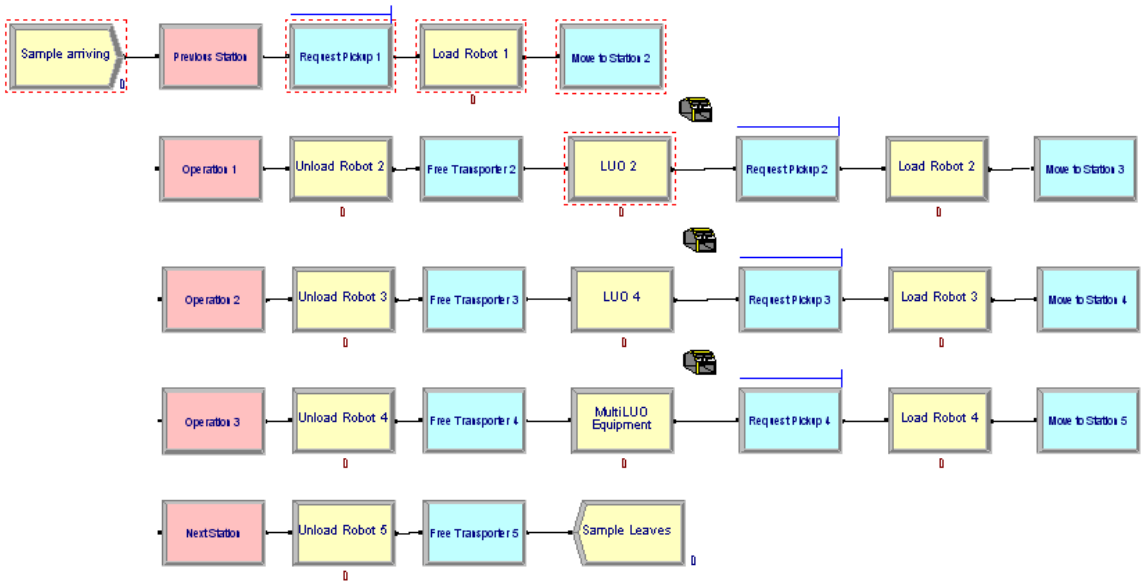

Figure 6.15 Integration of Centro LB 960 Microplate Luminometer

The Arena® Simulation validates the approach.

This example helps to validate also the Automation Modules. When defining them, only equipment 2 where taking into consideration, but this shows that the Automation Modules are valid and may be used also for equipment that has not been considered so far.

6.5.3. Discussion

Group 3 equipment (that is, complex systems that are capable of carrying out diverse tasks) may help reduce the automation line footprint. This might not necessarily be true as some of such “do all” solutions may integrate more operations than what is really needed to implement some given protocol.

As a result of replacing more than one instrument for only one, fewer instruments need to be integrated and controlled. The downside is that the instrument itself needs to be programmed, but many such systems are claimed to be easy to configure.

Timing considerations issues may also come up whenever introducing equipment that executes more than one operation. As already mentioned, the biotechnology protocols have very

specific timing conditions. For a reaction to occur, the required execution times have to be followed strictly. As explained earlier in this section, once a decision is made to use equipment from Group 3, the actual compound system equipment is viewed by the designer as a black box. The designer controls the instrument following the vendor's guidelines and specifications.

6.6. Additional Laboratory Operation

Some lab operations that have not been covered so far, such as restocking of labware or mid-process equipment change, that are used quiet frequently in automation of biotechnology processes have not for some reason been considered in the literature as part of the basic 12 LUO operations. Even though the inclusion of these operations does not mean the defining of new LUOs, the definitions of some of the already existing LUOs may need to be extended. Although such operations do not perform an actual action on a sample (like separation, agitation, homogenization, and so on), such operations are indispensable part of some protocols procedures. One may divide such operations into two types: manual operations and automated operations.

6.6.1. Manual Operations

Manual operations that may be needed in an automation line and that have not been included as LUOs in previous literature are: restocking of labware dispensers, replenishment of reagent dispensers and mid-process equipment interchange.

6.6.1.1. Restocking of labware dispensers and replenishment of reagents

Labware dispensers as well as liquid dispensing reservoirs have limited capacities. For example, the DNA Purification System by Thermo Scientific involves a carousel conveyor with capacity of 120 plates. Likewise, any liquid dispenser unit has a maximum volume to hold reagents. These given capacities may not be sufficient for the automation line shift period. In other words, the system might be able to process more samples in a given period of time if that causes an exceeding of labware available volume or the reagent volume that is available. When this happens, operators manually restock the labware or replenish the reagents, whichever is needed. Obviously for a smooth functioning of the system these operations need to be taken into consideration and need to be planned for ahead as done with any other operation.

As previously mentioned such restocking operations in many small-to-medium laboratories are done manually. However for example in compound library systems the process of reagent replenishment is automated. At the University of Cincinnati the Drug Discovery Center's Compound Library is a completely automated warehouse size laboratory. The system is capable of replenishing any of the 200,000 compounds available in the library by means of 3-D track-based manipulation system. The system is capable of fetching the desired compound to be placed in any of the digital dispensers of the High-Throughput Screening system (University of Cincinnati). Automation solutions that are equally sophisticated are available in most of the uHTS facilities.

As the automated restocking of labware dispensers requires manipulation and transportation of labware from storage to the system, such an operation may be included in LUO 1: Manipulation. In the case of replenishment of reagent dispensers there are two options: 1) the empty reservoir is exchanged for a full one, which too requires only manipulation and transportation of the reservoir and may be included in LUO 1. 2) The empty reservoir has to be filled up, which requires liquid handling operations and thus may be included in LUO 2: liquid handling. Which LUO to use is dictated by the liquid dispenser used depending whether or not it has removable reservoirs.

6.6.1.2. Mid-process equipment interchange

Mid-process equipment interchange is an operation that would only be necessary if the system is producing batches of different protocols or samples within a same shift and if such batch changes necessitate an equipment change. As with the restocking of labware dispensers, this operation is currently carried out manually by an operator, but automated solutions might be available in the future. As it involves the manipulation of actual equipment, this operation may be included in LUO 1, but keeping in mind that the description of LUO 1 has to be extended to include equipment manipulation.

6.6.2. Automated Operations

Automated operations that have not been yet included in the LUOs and are essential for full description of the protocols in terms of LUOs and its subsequent complete automation are

labeling operations and monitoring operations. These operations have been considered as operations for biotechnology automation, but they have not been described within the basic LUOs context.

6.6.2.1. Labeling operations

As mentioned several time in the previous chapters, labeling steps are left for future research. Their use in biotechnology is briefly described next.

Labeling operations are required if the designer attempts equipment sharing (other than the transportation and manipulation infrastructure) throughout the production line. Whenever automation equipment (such as a centrifuge) is shared among multiple protocol steps parts need to be labeled to allow the management system to keep track of them. Labeling requires two actions: reading a label off the sample or writing a label to be attached to a part. Each of the operations may require a different station. Only one label writing station would be necessary at the beginning of the process to attach a label to each of the parts. For the reading of the labels, multiple reading stations are needed throughout the system to track the samples all times. As labeling parts is done for tracking and managing parts, these operations may be included in LUO 12: Documentation. In this case, it would be more suitable to redefine LUO 12 to encompass all system managements operations including the management of the data documentation for analysis and reports and the traceability of each of the parts, and thus renaming it system management operations. The inclusion of labeling allows for automation lines that may no longer be sequential but rather feature multiple branching nodes.

6.6.2.2. Monitoring

Although monitoring of samples may already be included in LUO 10: measurements and direct detection, it is worth making the distinction between “measurement and direct detection”, and “monitoring”. Measurement and direct detection are basically downstream operations. It is considered measurement and direct detection whenever the aim of the process is to measure a given characteristic of the sample and the operation is executed on every sample. Monitoring is different and is more commonly used in upstream protocols. It uses measurements and direct detection but the goal is to make sure that the process is working well, hence monitoring. As most

of these detection methods involve the addition of foreign reagents to the sample, the operation is only carried out to a small sample of the samples. Therefore, monitoring may be included in LUO 10 making the distinction that when monitoring operations are involved they are not carried out in every single sample.

7. STRATEGIES FOR QUEUE CLEARING AND PARALLELIZATION IN BIOTECHNOLOGY AUTOMATION DESIGN

7.1. Introduction

In many respects the automation line design process for biotechnology applications may resemble those done in mechanical or electronic assembly applications; however there are some features that may be unique to biotechnology. Biotechnology processes often involve important constraints that have to be taken into consideration due to the nature of the biological products. Many biotechnology protocol steps often have strict timing and environmental conditions that must be followed in order to obtain the highest possible yield and product quality. In addition, the samples that flow through the automation line stations may have a relatively short shelf time. Thus the samples must be processed within the given time constraints or the product may be in risk of degradation. In any production line bottlenecks in the form of queues form whenever the parts input rate becomes larger than the rates that are compatible with the service time of each station. Such queues once formed tend to grow and without special automation line control measures may exceed a critical size that is related to the permitted and biologically dictated shelf time limits.

This chapter discusses several automation line design issues that are common to almost all types of manufacturing automation processes. These may arise from the following questions:

- How should we select the system input rate?
- When do queues begin to form?
- How do we limit the size of specific queues?
- What can be done to cap the queue waiting time to be below specific time limits?
- What is the throughput of the system?

Two design strategies to mitigate potential problems of growing queues are proposed. The first approach is a Queue Clearing Strategy via system input control. For any system input rate and the set of service times of each production line station, it is possible to assess if queues form anywhere in the system. There exists a critical input rate below which no queues form. A queue clearing strategy is proposed, to be deployed whenever the input rate exceeds a known critical value. This queue clearing strategy uses information about the maximum allowed shelf time in each queue (and/or station). Such a control method requires periodic stopping of inputting new raw material into the automation line where the time between stopping of the input and resuming it is the time needed to clear excess parts from the queues in order to maintain the queues within allowable sizes. Automation line designers are then able to study the effect of any high input rate (and its related clearing strategy) on the throughput of the process.

A second approach involves the parallelizing of select critical stations. An algorithm is developed to assist the automation line designer determine how many parallel stations need to be implemented at certain slow or critical steps of the process in order to avoid the formation of queues at any input rate higher than the critical rate known for a single sequential line. This approach may also allow an increase of the process throughput to any desired level within given cost constraints. System throughput enhancement via parallelizing of stations may of course also be implemented for input arrival rate that is faster than the critical arrival rate of the system.

7.2. Validation Tool: MATLAB® Simulator

A MATLAB® simulator was developed to evaluate the queue clearing and parallelization strategies formulated in this chapter. The MATLAB® simulator developed mimics the basic transfers of the parts from one station to the next by actually counting the parts as these moves throughout the system. The simulator computes the performance of the system throughout a shifts period of time. It shows the progression of the parts and the queues for each of the stations.

The MATLAB® simulator was developed under the assumption that the transportation time from one station to the next is much smaller that the time that takes each station to process the parts and therefore it was neglected. As soon as a sample is processed by a station it

becomes available for the next station. This assumption is used throughout this entire chapter and will be further discussed in a latter section.

Other than the same timing assumptions, the simulator does not use any of the formulas developed in this dissertation and in particular in this chapter.

The MATLAB® simulator was validated by comparing the results obtained from the MATLAB® simulator to the ones obtained with the Arena® simulation software for various models. As the results for the MATLAB® simulator were within acceptable margins (± 2% of total throughput for the systems) it was established that the neglecting of transportation times in the case of multi-manipulator implementation is justified. For a more detailed explanation of the MATLAB® simulator and its code see Appendices B and C.

The MATLAB® simulator is employed to simulate examples, and validate the methodologies and design formulas formulated throughout this chapter.

7.3. Definitions, Assumptions, and Basic Relationships

A sequential automation line consisting of P consecutive stations, from Station $p = 1$ to Station $p = P$ is considered. It is assumed that a number of raw samples (often referred to in this chapter as "parts"), NP, enter the automation system at Station 1 at a constant rate of AR. AR is the system Arrival Rate. Let $AP = 1/AR$ denote the arrival period of parts into the system. Similarly, let $Ap(p)$ be the parts arrival period to a station p where $p = 1, 2, \ldots, P$. Thus, $Ap(1) = AP$.

The starting time of the automation line at the beginning of each shift is denoted as $t = 0$. The shift ends at $t = tshift$ (For instance, a typical shift may have $tshift$ = 480 minutes) at which the automation line is shut down. The shut down stage is needed to assure that no parts stay inside the system after the end of the shift. In biotechnology automation lines, due to the biological nature of the parts, any samples that remain in the production line past the shift end must be discarded.

Each station is characterized by at least four parameters, two of which are related to physical and functional characteristics of the equipment and two are fundamental to the process. These are an integer number of parts, $Np(p)$, that Station p must process simultaneously; a

limited physical space where parts wait in queue to be serviced denoted as *MQs(p)* the maximum queue size; the service time, *St(p),* that takes Station *p* to process these *Np(p)* parts; and a maximum allowed shelf time, *MQt(p),* for parts waiting in the queue of Station *p* (Figure 7.1). For example, a common lab centrifuge may process six tubes arranged in radial symmetry (*Np(p) = 6*). Another example in automation systems for biotechnology involves centrifuges that processes two micro-arrays placed 180° one from another (*Np(p) = 2*). Other parameters that characterize a station, such as arrival period and turn on time, depend on a combination of some of the above station parameters and parameters of previous stations in the system.

Figure 7.1 shows a conceptual block diagram of Station *p*. The equipment of the station is represented by the red box where each slot, 1 through *Np(p)*, represents one part. The green box represents the queue for the station that only takes a finite number of parts *MQs(p)*.

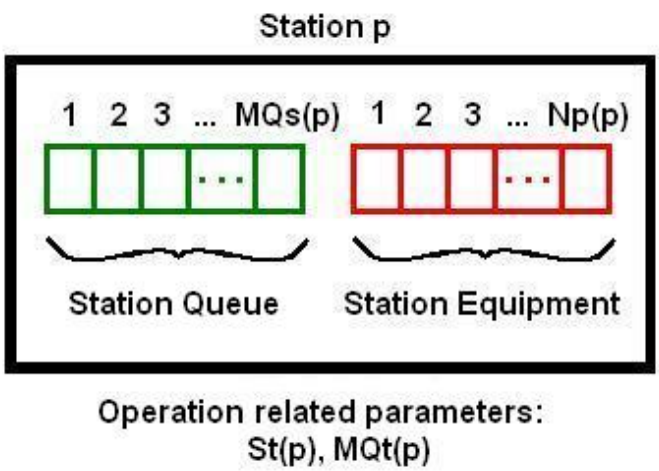

Figure 7.1 Conceptual block diagram of Station *p*

For simplicity of the derivations of the two queue clearing strategies, it is assumed throughout this chapter that the transport time from one station to the next is much smaller than the service time of either station. The inclusion of nonzero transport times is left for future extension of the results shown in this chapter. Thus, as Station *p* finishes the processing of its parts, all its output parts are assumed to be transferred *instantly* to Station *p + 1*. This means that the number of parts that are transferred to station *p* at each arrival period equal exactly to *Np(p – 1).*

Assuming initially that all stations have *Np(p) = 1*, and due to the zero transportation time assumption, once the production line reaches a steady state, each part arrives at each station periodically every *Ap(p)*. *Ap(p)* is the arrival period to Station *p* for *p > 1*, and it is equal to either *Ap(p - 1)* or to *St(p - 1)*, whichever is larger:

$$\begin{aligned} &Ap(1) = AP \\ &Ap(p) = \max(Ap(p-1), St(p-1)),\ p = 2,...P \end{aligned} \tag{7.1}$$

Figure 7.2 illustrates a sequential automation line block diagram. The system has *P* stations and the diagram shows the basic parameters associated with each station - *St(p)*, *Ap(p)*, and *Np(p)*. As in Figure 7.1, the red box and green box represent respectively the station processing equipment and queue. The diagram further illustrates how arrival periods propagate. This interaction will be further explained below.

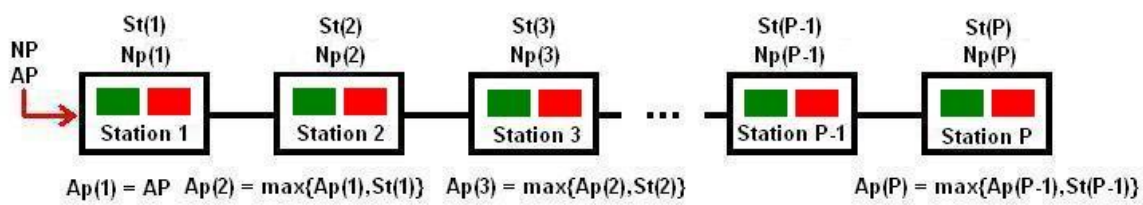

Figure 7.2 Conceptual block diagram for an automation line

As *Ap(p)* always takes the largest value between *Ap(p – 1)* and *St(p - 1)*, the largest *St(p)* which corresponds to the slowest station (for the case of *Np(p) = 1* for all stations) propagates throughout the system. *Ap(p)* is therefore a non-decreasing function of *p* as illustrated in the following example.

Example 7.1: let a production line have 3 stations and an arrival period of *AP* = 10 (time units). The service times for each of the stations are given as: *St(1)* = 20, *St(2)* = 10, and *St(3)* = 15. It is assumed that each station can process a single sample at a time.

The propagation of the parts throughout the system for Example 7.1 is shown in Table 7.1. In Table 7.1, *NIN* is the number of parts in the system, *Time* is the unit-time at which a part enters the system or moves from one station to the next. The base-time taken in Example 7.1 is 5. *Nin(p,t)* is the number of parts that enter a given station, and *Nout(p,t)* is the number of parts that leave (processed) a given station.

For easier visualization of the results, in Table 7.1 *Ap(p)* appears highlighted in yellow (every other period) and *St(p)* is highlighted in blue (every other period when appropriate) for each of the stations. Every other new part arriving into the system *NIN* is highlighting in grey.

Table 7.1 Results for Example 7.1

AP = 10		*Station 1, St(1) = 20*		*Station 2, St(2) = 10*		*Station 3, St(3) = 15*	
NIN	*Time*	*Nin(1,t)*	*Nout(1,t)*	*Nin(2,t)*	*Nout(2,t)*	*Nin(3,t)*	*Nout(3,t)*
1	0	1	0	0	0	0	0
1	5	1	0	0	0	0	0
2	10	2	0	0	0	0	0
2	15	2	0	0	0	0	0
3	20	3	1	1	0	0	0
3	25	3	1	1	0	0	0
4	30	4	1	1	1	1	0
4	35	4	1	1	1	1	0
5	40	5	2	2	1	1	0
5	45	5	2	2	1	1	1
6	50	6	2	2	2	2	1
6	55	6	2	2	2	2	1
7	60	7	3	3	2	2	1
7	65	7	3	3	2	2	2
8	70	8	3	3	3	3	2
8	75	8	3	3	3	3	2
9	80	9	4	4	3	3	2
9	85	9	4	4	3	3	3

In Example 7.1 therefore:

$$P = 3$$
$$Ap(1) = AP$$
$$\rightarrow Ap(1) = 10$$
$$Ap(p) = \max(Ap(p-1), St(p-1)), p = 2,3$$
$$\rightarrow Ap(2) = \max(Ap(1), St(1)) = \max(10, 20) = 20$$
$$\rightarrow Ap(3) = \max(Ap(2), St(2)) = \max(20, 10) = 20$$

Figure 7.3 shows the simulation results for a longer period of time for the system in Example 7.1. It shows the number of parts that have come into the station at the given times. *Ap(1)*, *Ap(2)*, and *Ap(3)* are easily recognizable and are shown in the graph. The periodicity of the parts entering each of the stations can be verified in Figure 7.3.

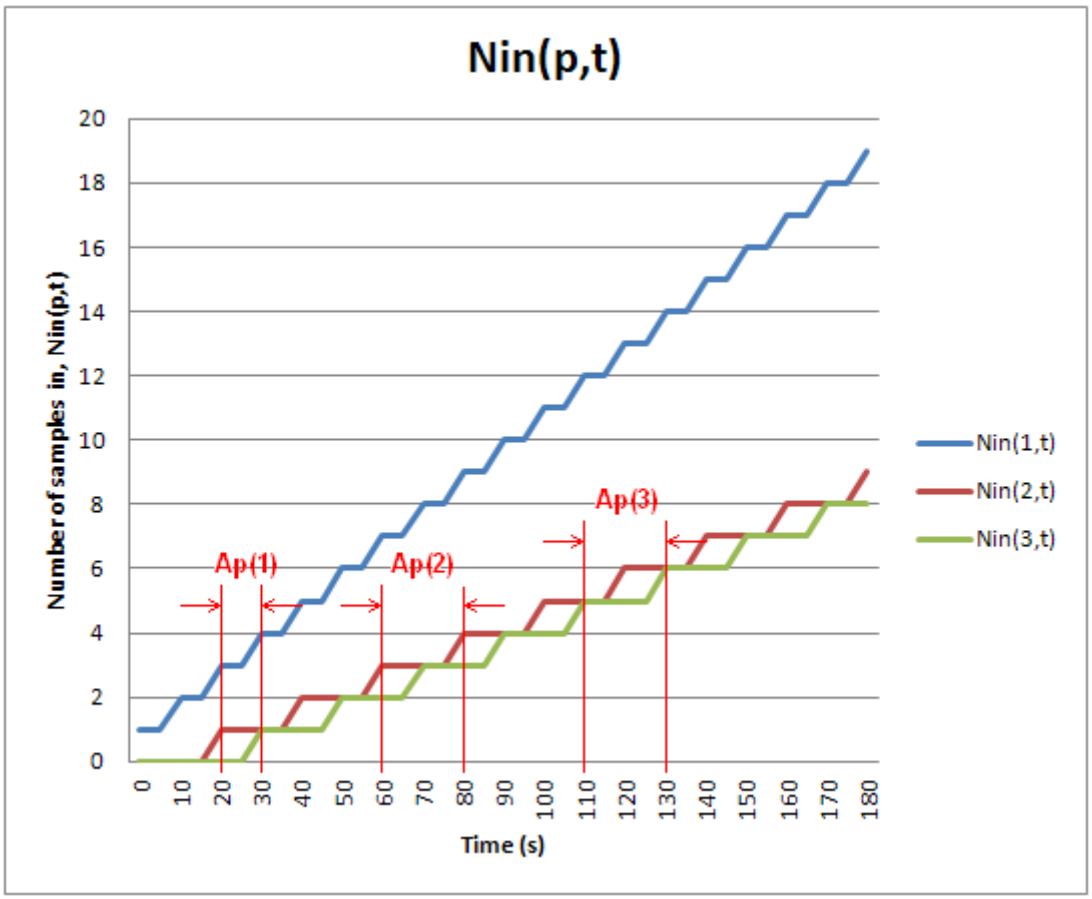

Figure 7.3 *Nin(p,t)* for Example 7.1

In general as one or more of the stations may process more than one part at a time, arrival period to each station may fluctuate. The incoming parts to any given station may in general arrive pseudo-randomly, at some long but deterministic period. The identification of such period allows the prediction of other system properties.

Example 7.2: consider the following five station system: *AP = 10s* seconds, Station *1*: *St(1) = 10s*, *Np(1) = 1*, Station *2*: *St(2) = 40s*, *Np(2) = 3*, Station *3*: *St(3) = 10s*, *Np(3) = 5*, Station *4*: *St(4) = 10s*, *Np(4) = 2*, and Station *5*: *St(5) = 10s*, *Np(5) = 1*. The results for Example 7.2 are presented in Figure 7.4

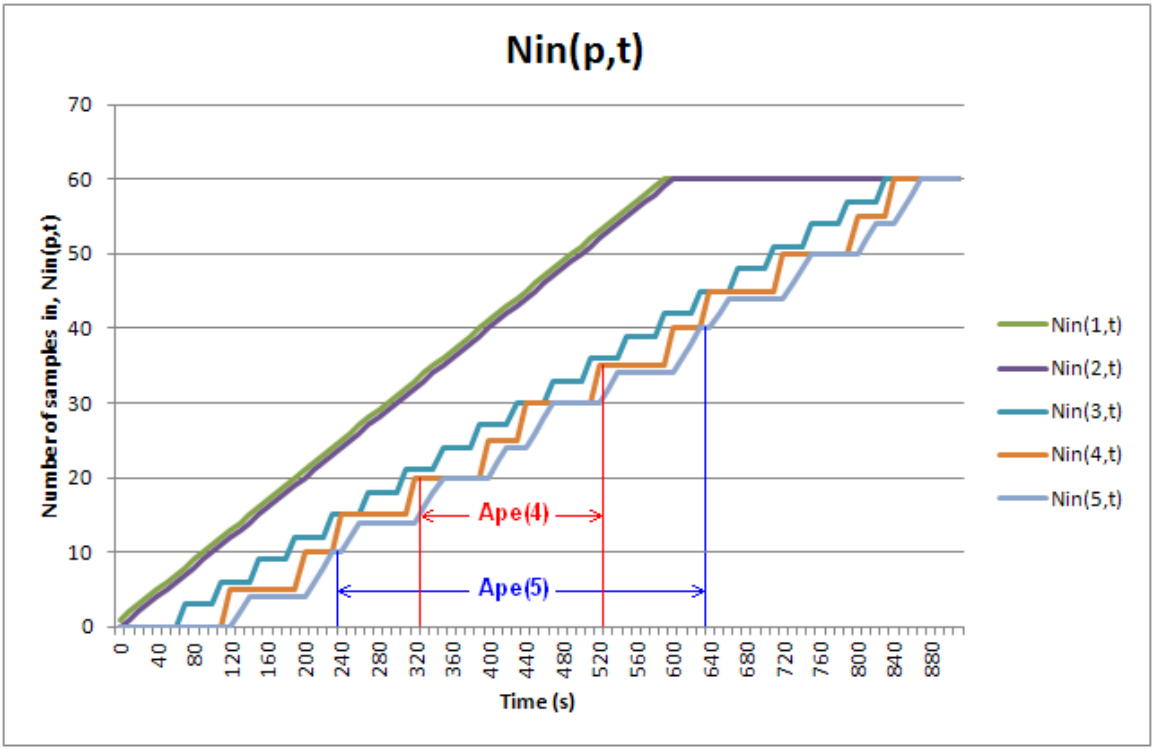

Figure 7.4 *Nin(p,t)* for a five station system

Figure 7.4 shows the *Nin(p,t)* values for the five stations in the system described above. *As NP = Np(1) = 1* for the system, the parts enter Station *1* and Station *2* with a period that can be calculated using (7.1). For Station *3*, as the *Np* values of the previous stations are integer multiples of one another (*1, 1, and 3*), *Np(2) = 3* parts arrive at Station *3* periodically. Station *4* and Station *5* illustrate the pseudo-randomness of incoming parts. For Station *4*, the parts arriving times at Station *4* may appear random. What is happening is that as Station *2* processes three parts at a time, and Station *3* processes five parts at a time, Station *3* needs to wait for more than one batch to be processed by Station *2* in order to start servicing its own parts. Once these parts are processed by Station *3* the output is passed along to Station *4*. The waiting time for parts to be available for service varies until the least common multiple of the previous *Np* values has entered Station *4* and thereafter the cycle starts again. For Station *4* it can be observed that every *200* seconds there are *15* parts which enter the unit. Likewise, for Station *5* it is observed that every *400* seconds there are *30* parts that enter the system.

Identifying the periodic cycle helps predict how and when parts move from one station to the next within a reasonable error margins. The error is never higher that *Np(p - 1)*. As a steady

state pattern is reached the error in this example seems to be no higher than ±1%. A trend line can be drawn to illustrate the predicted performance for *Nin(p,t)*. As an example Figure 7.5 shows *Nin(4,t)* and the trend line for the same station.

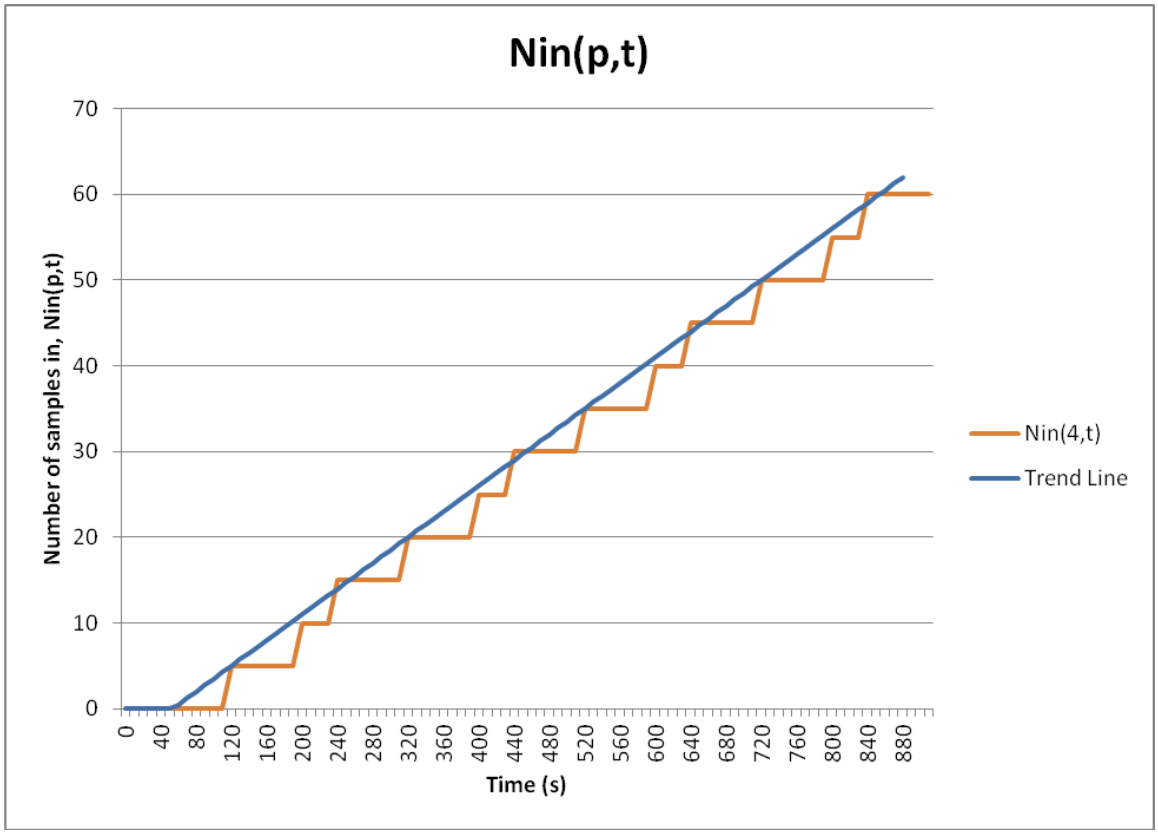

Figure 7.5 *Nin(4,t)* and trend line

To describe the relationship between *time* and the input of a station, two new parameters for each station are defined: the effective service time *Ste(p)* and the effective arrival period *Ape(p)*. The effective service time, *Ste(p),* is the actual service time of a station that processes multiple parts simultaneously. *Ste(p)* is calculated by dividing the real service time of the Station *p* *St(p)* by the number of parts the station processes at the same time *Np(p)*. Notice that although *Spe(p)* is defined for stations having *Np(p) > 1*, it is also valid for stations with *Np(p) = 1*. If the system has at least one station that has *Np(p) > 1*, then *Ste(p)* should be used for the calculations.

$$Ste(p) = \frac{St(p)}{Np(p)} \tag{7.2}$$

Similarly, the effective arrival period of the station is the perceived arrival period due to a upstream Station *p* having an *Np(p) > 1*. Likewise, if at least one station features *Np(p) > 1*, then *Ape(p)* should be used for all the calculations.

$$\begin{aligned} Ape(1) &= \frac{AP}{NP} \\ Ape(p) &= \max\big(Ape(p-1), Ste(p-1)\big),\ p = 2,...P \end{aligned} \tag{7.3}$$

(7.3) generalizes (7.1). We shall continue the derivations in this manner – simple formulas first for the case that each station processes only a single part, followed by a generalization to the case in which at least one of the units processes multiple parts.

Initially the trend line drawn to illustrate the predicted performance for *Nin(p,t)*, was calculated by finding the equation of the straight line that passes through the identified periodic points. Now, after defining the effective arrival period of a station *Ape(p)*, we can say that the trend line is the linearized version of *Nin(p,t)* with slope *1/Ape(p)*.

Assuming initially that all stations have *Np(p) = 1*, a station turns on whenever the first part arrives at the station at a time of *t = ton(p)*. For *Station 1,* the first part arrives at *ton(1) = 0.* At *t = 0,* all other stations are idle. As the first part is processed and moves from station to station, it turns each station on at *t = ton(p)*, which is the sum of the service times of all previous *p - 1* stations.

$$\begin{aligned} ton(1) &= 0 \\ ton(p) &= \sum_{i=1}^{p-1} St(i),\ p = 2,...P \end{aligned} \tag{7.4}$$

Figure 7.6 shows the timings associated with the first sample propagating through the stations turning them on one after the other.

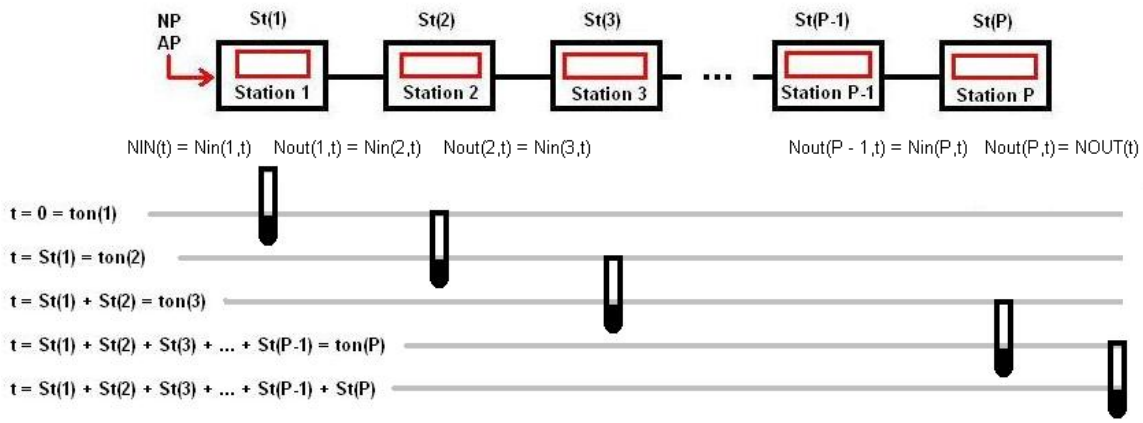

Figure 7.6 Propagation of the first sample for Example 7.1

In Example 7.1, *ton* for each of the stations is the time at which the first part arrives at the given station or equivalently whenever *Nin(p) = 1*, hence: *ton(1)* = 0, *ton(2)* = 20, and *ton(3)* = 30. By (7.4):

$$P = 3$$
$$ton(1) = 0$$
$$\rightarrow ton(1) = 0$$
$$ton(p) = \sum_{i=1}^{p-1} St(i),\ p = 2,3$$
$$\rightarrow ton(2) = St(1) = 20$$
$$\rightarrow ton(3) = St(1) + St(2) = 20 + 10 = 30$$

Clearly *ton(p)* is too a non-decreasing function of *p*.

To generalize *ton(p)* for the case where each *Np(p)* may take any positive integer value, the arrival period of the station and the number of parts processed by the previous station and the current station, both need to be considered in order to calculate the time when each station turns on. There are two cases: case 1, if the number of parts processed by a given station is less than the number of parts processed by the previous station (that is $1 \le Np(p) \le Np(p-1)$); and case 2, if the number of parts processed by a given station is greater than the number of parts processed by the previous station (that is $1 \le Np(p-1) < Np(p)$).

In case 1, *if* $1 \le Np(p) \le Np(p-1)$, as the number of parts entering the station for the first time, $Np(p-1)$, is greater than the number of parts that Station *p* can process, *Np(p)*. Station *p* turns on as it can start processing the arriving parts right away. The excess number of parts

arriving at Station *p* goes to the queue of Station *p*, and wait for the equipment next processing round. In other words, *ton(p)* for this case could be found from (7.4).

For case 2, if $1 \leq Np(p-1) < Np(p)$, however, the number of parts entering Station *p* for the first time is less than the number of parts Station *p* should process. For example, for a station with an $Np(p) = 2$, one part will not turn on the station; the station has to wait for a second part before it can start processing. A second part arriving *Ap(p)* time-units later than the first sample, turns on the station. Due to this behavior a new variable *tfirst* is introduced to the system to define the time at which the first part or parts enters the system.

To calculate exactly how many *Ap(p)* time-units later on Station *p* needs to turn-on, the ratio between *Np(p)* and *Np(p - 1)* must be included in the *ton(p)* calculation. Also, as those extra waiting times for those stations where $1 \leq Np(p-1) < Np(p)$ have to be included into downstream calculations, for each station we first calculate *tfirst(p)* and then we calculate *ton(p)* before calculating the next station's turn on time.

The time *tfirst(1)* is by definition 0 as the first part or parts enter the system at $t = 0$ as explained before. For all other station, the first part will arrive to the system one service time after the previous station has been turned on. Hence,

$$\begin{aligned} &tfirst(1) = 0 \\ &tfirst(p) = ton(p-1) + St(p-1),\ p = 2,\ldots,P \end{aligned} \tag{7.5}$$

Calculating *tfirst(p)* for Station *p* first, allows to compute *ton(p)* by adding the extra arrival periods needed for the station to receive all necessary *Np(p)* to the time *tfist(p)*. The expression for *ton(p)* is thus generalized as follows:

$$\begin{aligned} &ton(1) = tfirst(1) + \left(\left\lceil \frac{Np(1)}{NP} \right\rceil - 1 \right) \cdot AP \\ &ton(p) = tfirst(p) + \left(\left\lceil \frac{Np(p)}{Np(p-1)} \right\rceil - 1 \right) \cdot \max\big(St(p-1), Np(p-1) \cdot St(p-2)\big),\ p = 2,\ldots P \end{aligned} \tag{7.6}$$

The square brackets in (7.6) are the notation for the **ceiling** function. The ceiling function is defined as: $\lceil x \rceil = \min\{n \in Z | n \geq x\}$, where *x* is a real number, *n* is an integer, and *Z* is the set of integers. Likewise, the notation for the **floor** function is $\lfloor x \rfloor$, and it is defined as:

$\lfloor x \rfloor = \max\{m \in Z | m \leq x\}$. The "ceiling" function finds the "least integer upper bound" whereas the "floor" function finds the "largest integer lower bound".

(7.6) is illustrated in Example 7.3.

Example 7.3: Consider the system discussed in Example 7.1, in which each station processes a single part, except for *Np(2)* = 2.

Table 7.2 Results for Example 7.3

AP = 10		*Station 1, St(1) = 20, Np(1) =1*		*Station 2, St(2) = 10, Np(2) = 2*		*Station 3, St(3) = 15, Np(3) = 1*	
NIN	*Time*	*Nin(t,1)*	*Nout(t,1)*	*Nin(t,2)*	*Nout(t,2)*	*Nin(t,3)*	*Nout(t,3)*
1	0	**1**	0	0	0	0	0
1	5	1	0	0	0	0	0
2	10	2	0	0	0	0	0
2	15	2	0	0	0	0	0
3	20	3	1	**1**	0	0	0
3	25	3	1	1	0	0	0
4	30	4	1	1	0	0	0
4	35	4	1	1	0	0	0
5	40	5	2	**2**	0	0	0
5	45	5	2	2	0	0	0
6	50	6	2	2	2	**2**	0
6	55	6	2	2	2	2	0
7	60	7	3	3	2	2	0
7	65	7	3	3	2	2	1
8	70	8	3	3	2	3	1
8	75	8	3	3	2	3	1
9	80	9	4	4	2	3	2
9	85	9	4	4	2	3	2

Table 7.2: Results for Example 7.3 and it has initially the same notation as Table 7.1. Green highlights, bold letters, and color numbers were added to show the times and inputs of interest. A black bold number represents *tfirst(p)*, a blue bold number represents *ton(p)*, and a red bold number is used for those stations were *tfirst(p) = ton(p)*.

Table 7.2 includes *tfirst(p)* and *ton(p)* for the three stations in Example 7.3, which are *tfirst(p) = 0*, *ton(1)* = 0, *tfirst(2) = 20*, *ton(2)* = 40, *tfirst(p) = 50*, and *ton(3)* = 50.

By (7.5) and (7.6):

$$tfirst(1) = 0$$

$$ton(1) = tfirst(1) + \left(\left\lceil \frac{Np(1)}{NP} \right\rceil - 1 \right) \cdot AP$$

$$\rightarrow ton(1) = 0 + \left(\left\lceil \frac{1}{1} \right\rceil - 1 \right) \cdot 10 = 0$$

$$tfirst(p) = ton(p-1) + St(p-1)$$

$$ton(p) = tfirst(p) + \left(\left\lceil \frac{Np(p)}{Np(p-1)} \right\rceil - 1 \right) \cdot \max\left(St(p-1), Np(p-1) \cdot St(p-2) \right)$$

$$\rightarrow tfirst(2) = ton(1) + St(1) = 0 + 20 = 20$$

$$\rightarrow ton(2) = tfirst(2) + \left(\left\lceil \frac{Np(2)}{Np(1)} \right\rceil - 1 \right) \cdot St(1) = 20 + 1 \cdot 20 = 40$$

$$\rightarrow tfirst(3) = ton(2) + St(2) = 40 + 10 = 50$$

$$\rightarrow ton(3) = tfirst(3) + \left(\left\lceil \frac{Np(3)}{Np(2)} \right\rceil - 1 \right) \cdot Np(2) \cdot St(1) = 50 + 0 \cdot 2 \cdot 40 = 50$$

Two other times of interest are also defined: *tlast(p)*, the time at which the last part enters the system; and *toff(p)*, the time at which the last part has been processed by the station and the station is turned off. These two times are further explained in a later section. For now it is important to be aware that such times exist.

7.4. Queue Size Computation

For *Np(p) = 1*, a part that arrives at any station that may be busy processing other parts at that time is placed in a queue that is available at this station. A queue in Station *p* forms whenever *St(p) > Ap(p)*. If *Ap(p) > St(p)* for every *p = 1,2, …, P,* then no queues ever form throughout the system. Otherwise queues may form at some of the stations in the system. Let us assume that if a part arrives from Station *p - 1* at Station *p* at a time *t* at which the queue of Station *p* has at least one part waiting, this arriving part is then placed at the end of the queue. As soon as Station *p* becomes available to process a new part, it accepts the first part waiting in its queue. At any time *t* and Station *p, Qs(p,t)* denotes the size of the queue of Station *p*. In the more general case of *Np(p) > 1*, the relationships that matter are *Ape(p) > Ste(p)* or *Ste(p) > Ape(p)*.

For *Np(p) > 1*, parts arriving at the stations are placed inside the processing unit until it is full and ready to start processing. The number of parts inside the equipment processing unit at any given time are denoted by N_{SPU} *(p,t)* (Number of samples inside the processing unit), which may take values from *0* to *Np(p)*. Trivially $N_{SPU}(p,t) = 1$ for *Np(p) = 1*. Overflow parts are placed outside the processing unit in the station queue.

At any Station *p,* if *Ape(p) < Ste(p),* a queue begins to form. Station *p* queue size *Qs(p,t)* equals the difference between *Nin(p,t)*, the total number of all parts that have entered Station *p* between *tfirst(p)* and *t* (a time of interest), the total number of all the parts *Nout(p,t)* that have exited the station during the interval between *ton(p)* and *t*, and $N_{SPU}(p,t)$, the parts that are being processed, or still accumulate to be processed.

$$Qs(p,t) = Nin(p,t) - Nout(p,t) - N_{SPU}(p,t) \tag{7.7}$$

Example 7.4: A follow up of Example 7.1.

Table 7.3 Results for Example 7.4

AP = 10		Station 1, St(1) = 20, Np(1) = 1				Station 2, St(2) = 10, Np(2) = 1				Station 3, St(3) = 15, Np(3) = 1			
NIN	Time	Nin (t,1)	Qs (t,1)	NSPU (1)	Nout (t,1)	Nin (t,2)	Qs (t,2)	NSPU (2)	Nout (t,2)	Nin (t,3)	Qs (t,3)	NSPU (3)	Nout (t,3)
1	0	1	0	1	0	0	0	0	0	0	0	0	0
1	5	1	0	1	0	0	0	0	0	0	0	0	0
2	10	2	1	1	0	0	0	0	0	0	0	0	0
2	15	2	1	1	0	0	0	0	0	0	0	0	0
3	20	3	1	1	1	1	0	1	0	0	0	0	0
3	25	3	1	1	1	1	0	1	0	0	0	0	0
4	30	4	2	1	1	1	0	0	1	1	0	1	0
4	35	4	2	1	1	1	0	0	1	1	0	1	0
5	40	5	2	1	2	2	0	1	1	1	0	1	0
5	45	5	2	1	2	2	0	1	1	1	0	0	1
6	50	6	3	1	2	2	0	0	2	2	0	1	1
6	55	6	3	1	2	2	0	0	2	2	0	1	1
7	60	7	3	1	3	3	0	1	2	2	0	1	1
7	65	7	3	1	3	3	0	1	2	2	0	0	2
8	70	8	4	1	3	3	0	0	3	3	0	1	2
8	75	8	4	1	3	3	0	0	3	3	0	1	2
9	80	9	4	1	4	4	0	1	3	3	0	1	2
9	85	9	4	1	4	4	0	1	3	3	0	0	3

Table 7.3 is the same as Tables 7.1 and 7.2 with added the columns for *Qs(p,t)* and $N_{SPU}(p)$ for each station. Note that since Station *1* is the only station for which *Ape(p) < Ste(p)* (*Ape(1) = 10 < Ste(1) = 20*), a queue appears only in Station *1*. For the other stations no queues form.

For *Np(p) = 1*, the integer *Nin(p,t)* is calculated by dividing the total time that has passed from the initial time *ton(p)* by the arrival period of that given station and adding the number of parts processed by the previous station, which in this case is 1. The floor function is used as only integer values are valid for numbering parts. The number of incoming parts to Station *1* is calculated slightly differently as shown below.

$$\begin{aligned} Nin(1,t) &= \left\lfloor \frac{t}{AP} \right\rfloor + 1 \\ Nin(p,t) &= \left\lfloor \left(\frac{t - ton(p)}{Ap(p)} \right) \right\rfloor + 1,\ p = 2,..,P \end{aligned} \tag{7.8}$$

(7.8) is valid for *ton(p) ≤ t ≤ tlast(p)*. For *t ≤ 0 < ton(p)*, *Nin(p,t) = 0*. For *t > tlast*, *Nin(p,t) = Nin(p,tlast(p))*.

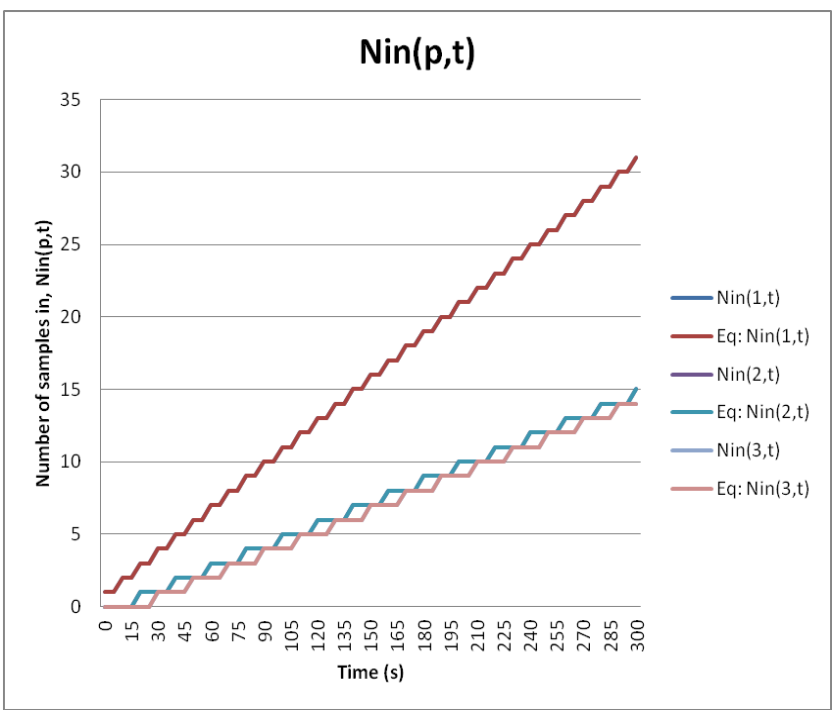

Figure 7.7 Results for *Nin(p,t)* simulated and calculated using (7.8) for Example 7.1

Figure 7.7 compares *Nin(p,t)* values computed by (7.8) to those computed by the MATLAB® simulator for Example 7.1.

For the general case *Np(p) > 1*, the integer *Nin(p,t)* is approximated by dividing the total time that has elapsed, from the time the first part entered the system *tfirst(p),* by the effective arrival period of that given station *Ape(p)*, and adding the number of parts processed by the previous station. Need to keep in mind that arrival of multiple parts is included in the calculations as *Ape(p)* replaces *Ap(p)*. To properly approximate *Nin(p,t)*, the modulo operation (mod) is included as in the general case only multiple numbers of *Np(p - 1)* are valid results. The expression to approximate the values of *Nin(p,t)* is:

$$Nin(p,t) \approx \left(\frac{t - tfirst(p)}{Ape(p)} \right) - \mathrm{mod}\left(\left(\frac{t - tfirst(p)}{Ape(p)} \right), Np(p-1) \right) + Np(p-1) \qquad (7.9)$$

(7.9) is valid for *tfirst(p) ≤ t ≤ tlast(p)*. For *t ≤ 0 < tfirst(p)*, *Nin(p,t) = 0*. For *t > tlast(p), Nin(p,t) = Nin(p,tlast(p))*

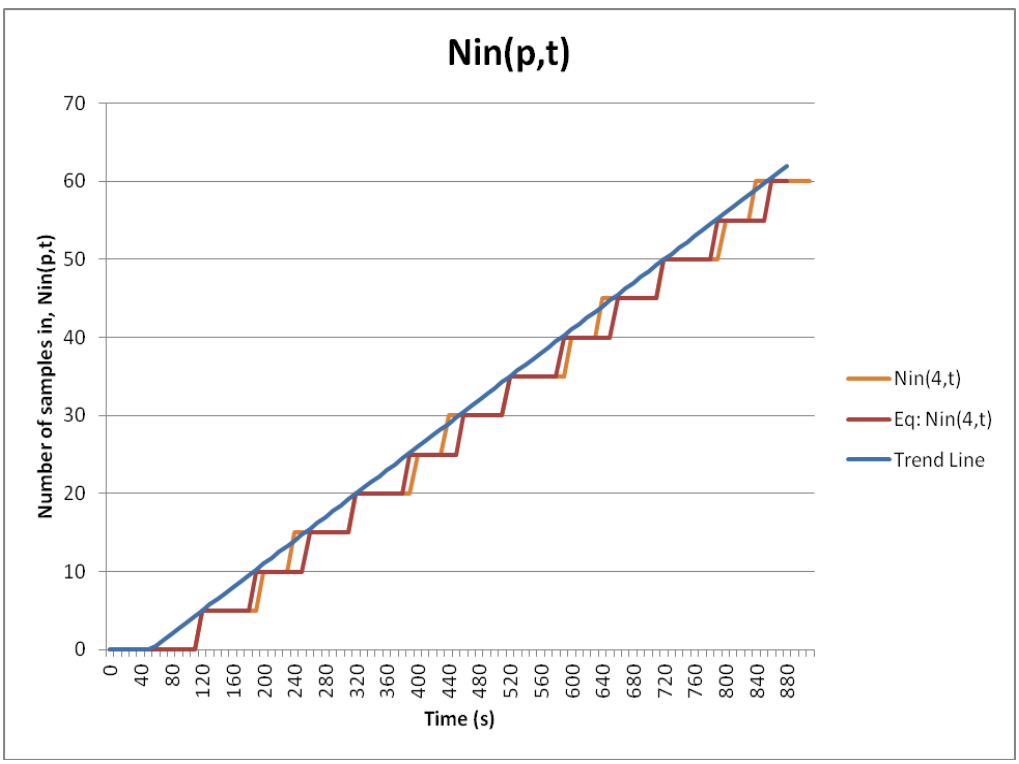

Figure 7.8 Results for *Nin(4,t)* simulated and calculated using (7.9) for Example 7.2 and predicted trend line

Figure 7.8 shows the results for Station *4* in Example 7.2. It compares the actual values for *Nin(4,t)* calculated via the MATLAB® simulator with those calculated using (7.9). It also includes the expected trend line for the parts entering Station *4*. As explained when Example 7.2 was first introduced, stations which have non-commensurate *Np* values with neighboring stations, may have periodic arrival times or have a periodic cycle with a longer period (depending on the relation between upstream *Nps*). Expression (7.9) approximates the behavior to a periodic one. This is an acceptable practical approximation as the error for the calculated value is never larger than the maximum *Np* of the system and both, calculated and simulated values, follows the same trend line.

For *Np(p) = 1*, there are two cases to take into account in order to calculate *Nout(p,t)* of Station *p*: 1) *Ap(p) < St(p),* or 2) *St(p) ≤ Ap(p)*.

We note that for queue size computation we use case 1 as that is the case when a queue indeed forms.

For *Ap(p) < St(p), Nout(p,t)* for Station *p* is calculated by dividing the time interval from the time that Station *p* has been turned on by its service time *St(p)*.

$$Nout(p,t) = \left\lfloor \left(\frac{t - ton(p)}{St(p)} \right) \right\rfloor \tag{7.10}$$

For *St(p) ≤ Ap(p)*, when no queue forms, *Nout(p,t)* for Station *p* is calculated by dividing the time interval since Station *p* has been turned on by its arrival period *Ap(p)*.

$$Nout(p,t) \approx \left\lfloor \left(\frac{t - ton(p)}{Ap(p)} \right) \right\rfloor \tag{7.11}$$

In general,

$$Nout(p,t) \approx \left\lfloor \left(\frac{t - ton(p)}{\max\{Ap(p), St(p)\}} \right) \right\rfloor \tag{7.12}$$

(7.10), (7.11), and (7.12) are only valid for *ton(p) ≤ t ≤ toff(p)*. By definition, for *0 ≤ t ≤ ton(p)*, *Nout(p,t) = 0* and for *t > toff(p)*, *Nout(p,t) = Nout(p,toff(p))*.

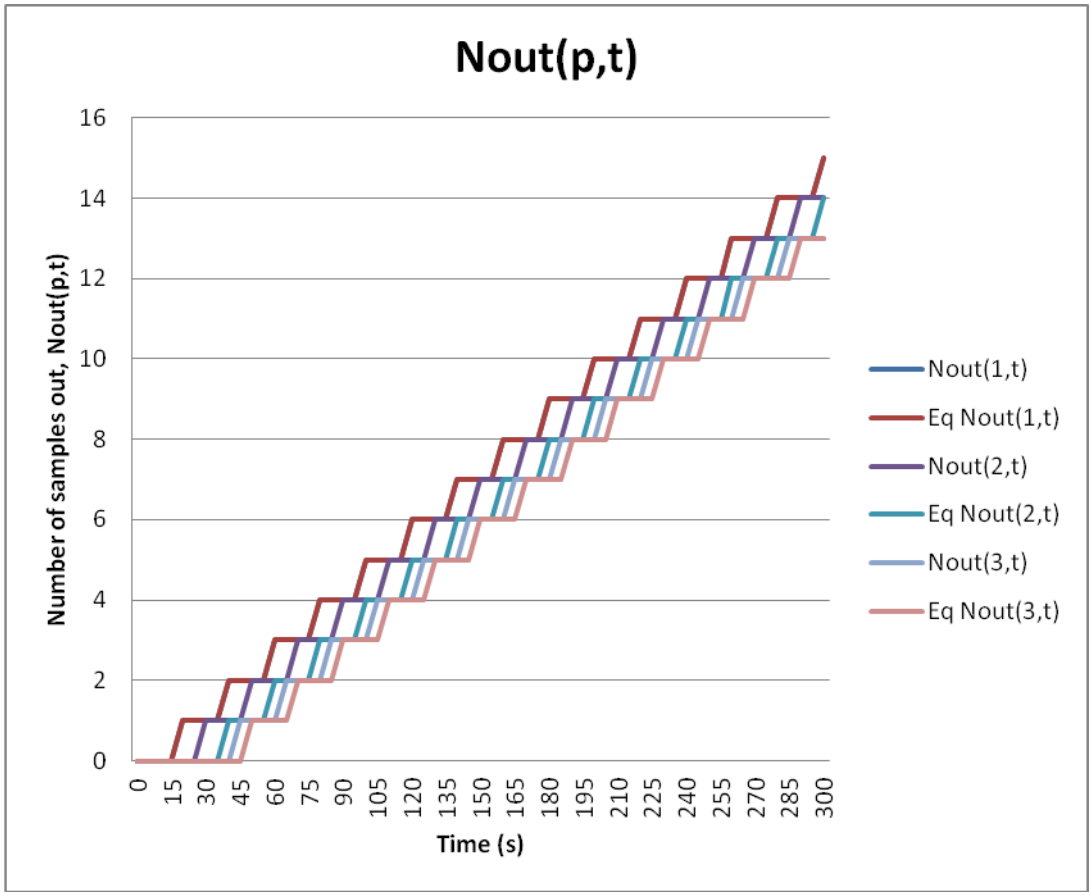

Figure 7.9 Results for *Nout(p,t)* simulated and calculated using (7.8)

Figure 7.9 helps us compare the results computed using (7.12) against the actual *Nout(p,t)* found by direct counting done by the MATLAB® simulator for Example 7.1. In this example both cases, *Ap(p)* < *St(p)* and *St(p)* < *Ap(p)*, are ilustrated. Figure 7.9 helps to visuallize two effects. First, for the case *Ap(p)* < *St(p)*, (7.12) presents no errors as the calculation is made using the time that the station actually takes to processes the sample *St(p)* (Station *1*). Second, for the case *St(p)* < *Ap(p)*, the expression introduces an error as it delays the values *Ap(p)* – *St(p)* units of time. When the error appear it will always be equal to *-1*. This happens as in the calculation the division is made by the value of *Ap(p)* (Station *2* and Station *3*). Although the expression is always correct at the end of the period, the expression does not predicts the correct time at which the parts enters the station. The exact time at which the sample enters could be predicted with a more complex calculation, or a multiple step calculation. This is not performed as the approximation obtained is acceptable for the case *St(p)* < *Ap(p)* as this is not the critical case.

The critical case is whenever queues are present, *Ap(p)* < *St(p)*, and for this case the expression has no errors. For stations where no queues are present, *St(p)* < *Ap(p)*, the expression predicts the number of parts at the end of each period. For other times the error will be equal to *-1* part.

For the general case *Np(p)* > *1*, the integer *Nout(p,t)* is approximated by dividing the total time that has passed from the time the station is turned on, *ton(p)*, by the maximum value between the effective arrival period *Ape(p)* the effective service time of the station *Ste(p)*. The arrival of multiple parts are taken into consideration in both *Ape(p)* and *Ste(p)*. The modulo operation (mod) is included as in the general case only multiple numbers of *Np(p)* are valid results. The expression to approximate the values of *Nout(p,t)* is:

$$Nout(p,t) \approx \left(\frac{t - ton(p)}{\max\{Ape(p), Ste(p)\}} \right) - \mathrm{mod}\left(\left(\frac{t - ton(p)}{\max\{Ape(p), Ste(p)\}} \right), Np(p) \right) \quad (7.13)$$

(7.13) is valid for *ton(p)* ≤ *t* ≤ *toff(p)*. By definition, for *0* < *t* < *ton(p)*, *Nout(p,t)* = *0* and for *t* > *toff(p)*, *Nout(p,t)* = *Nout(p, toff(p))*.

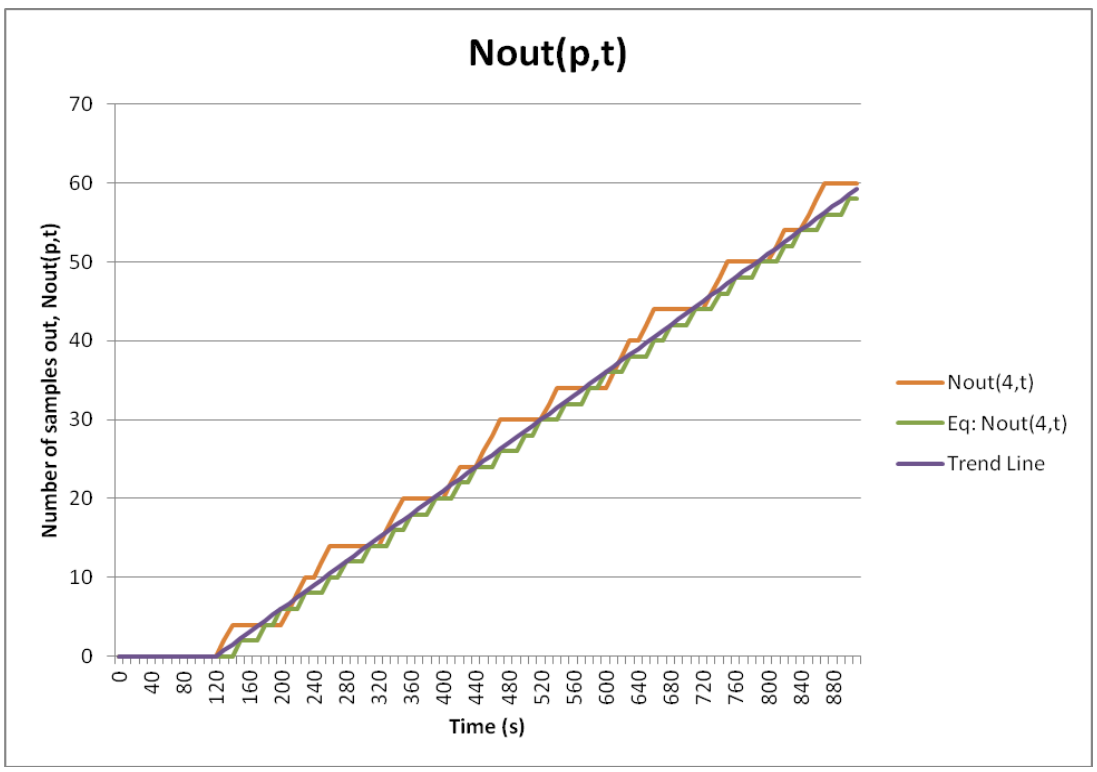

Figure 7.10 Results for *Nout(4,t)* simulated and calculated using (7.13) for Example 7.2 and predicted trend line

Figure 7.10 compares the results between the values calculated using (7.13) with those simulated with the developed MATLAB® simulator. It is clear that *Nout(p,t)* behaves similarly to *Nin(p,t)* for the case *Np(p) > 1*. This approximation is within a margin of the maximum *Np(p)* of the system.

Note that *Nin(p,t)* and *Nout(p,t)* for each Station *p* are both non-decreasing functions of *t*. Even though *Nin(p,t) = Nout(p - 1,t)*, both *Nin(p,t)* and *Nout(p,t)* expressions are to be used for computation of a queue size. The elimination of *Nout(p,t)* using *Nin(p - 1,t)* would create a recursive expression for *Qs(p,t)* that requires system wide computations. In comparison, the use of both in and out samples calculations allows a local calculation of queue size.

For *Np(p) = 1*, queues form whenever the processing unit of Station *p* is busy processing a prior sample. By substitution of (7.8) and (7.10) into (7.7):

$$Qs(p,t) = \left\lfloor \frac{t - ton(p)}{Ap(p)} \right\rfloor - \left\lfloor \frac{t - ton(p)}{St(p)} \right\rfloor \tag{7.14}$$

For *Np(p) > 1*, as queues form whenever the processing unit of Station *p* is busy processing a prior sample batch, it means that at any given time, there are *Np(p)* parts inside the processing unit whenever queues occur. This further implies that if a queue exists then $N_{SPU}(p,t) = Np(p)$. By substitution of (7.9) and (7.13) into (7.7):

$$\begin{aligned} Qs(p,t) \approx & \left(\frac{t - tfirst(p)}{Ape(p)} \right) - \mathrm{mod}\left(\left(\frac{t - tfirst(p)}{Ape(p)} \right), Np(p-1) \right) + Np(p-1) \\ & - \left(\left(\frac{t - ton(p)}{Ste(p)} \right) - \mathrm{mod}\left(\left(\frac{t - ton(p)}{Ste(p)} \right), Np(p) \right) \right) - Np(p) \end{aligned} \tag{7.15}$$

As a queue forms and parts are awaiting in the queue to be serviced, a waiting time (or queue time), *Qt(p,t)*, associated with *Qs(p,t)* may be defined as:

$$Qt(p,t) \approx \left\lceil \frac{Qs(p,t)}{Np(p)} \right\rceil \cdot St(p) \tag{7.16}$$

Qt(p,t) is calculated by dividing the number of samples waiting in line in the queue of Station *p* at a specific time, *Qs(p,t)*, by the number of units that Station *p* can process simultaneously, *Np(p)*. This finds the number of runs that Station *p* has to carry out in order to

process the parts in the queue starting at time *t*. The number of runs is then multiplied by the time that Station *p* takes to service each group of parts, *St(p)*.

From this point on, only the general case of multiple processed parts is addressed. Reduction to the case *Np(p) = 1* involves the obvious substitutions: *Ape(p) = Ap(p)*, *Ste(p) = St(p)*, *Np(p) = 1*, and *NP = 1*.

7.5. Throughput Calculations

The total number of samples *IN(t)* that have entered the system up to a given time instance *t* is calculated by dividing the time of interest by the input arrival period *AP* and then by multiplying the result by the number of parts *NP* that arrive at every interval.

$$IN(t) \approx NP \cdot \left\lfloor \frac{t}{AP} \right\rfloor + NP \tag{7.17}$$

The number of samples *OUT(t)* processed by the system up to a given time instance *t* is calculated by finding the number of processed units that have exited the last station, *p = P*. To calculate that number at any time of interest there is a need to divide the time that Station *P* has been on by the output period. Since *OUT(t)* is found by calculating the number of samples that exited the last station, the output period is the greater value between *Ape(P)* and *Ste(P)*.

$$OUT(t) \approx \left(\frac{t - ton(P)}{\max\{Ape(P), Ste(P)\}} \right) - \mathrm{mod}\left(\left(\frac{t - ton(P)}{\max\{Ape(P), Ste(P)\}} \right), Np(P) \right) \tag{7.18}$$

The *throughput, TP,* of the system is calculated by finding the number of samples processed by the system after the shift has ended. This *TP* is intrinsic to the given system. It is the maximum throughput a system with given *AP*, *NP*, *St(p)*, and *Np(p)* will be able to produce for a given *tshift*.

$$TP = OUT(tshift) \tag{7.19}$$

Taking a closer look at (7.18), the total output of the system at any given time, and therefore, at the end of a shift can be calculated in terms of the slowest station *SS* of the system. Calculating *Ape(P)* essentially propagates the greatest service time (*Ste(p)*) which belongs to the slowest station because it compares every service time for the previous stations and maintains the greatest value as in (7.3). A larger effective service time value indicates that the operation

that takes place in the given station, takes longer to process, and hence, it is a slower station. The station with the largest effective service time is therefore the slowest station. As (7.18) uses the maximum value between the *Ape(P)* and the *Ste(P)*, if a station other than the last station is the slowest station, this service time is reflected in the *Ape(P)* calculation, and this value is then used; on the other hand, if the slowest station is the last station *Ste(P)* is used for the output calculations.

If the effective arrival period *AP/NP* of the system equals the effective service time for the slowest station, *Ste(SS)*, the throughput value of the system is the maximum possible value as no queues are created. If smaller values than *AP/NP* for the effective arrival period are used, queues appear. If larger values than *AP/NP* are used, no queues are created but the throughput starts to decrease. The *AP* of the system is a design parameter that might not be easily modifiable. Some equipment such as incubators, storage units, or freezers may have fixed set input rate value.

For any automation line to be able to deliver a required throughput *TPr*, the specified throughput needs to be lower than the throughput of the system *TP* (*TPr* ≤ *TP*). Otherwise, modifications to the system, via different strategies such as parallelizing strategies, have to be made to be able to produce the required throughput.

7.6. System and Stations Shutdown

Biological samples tend to be fragile. Any sample that enters the system and is not fully processed by the end of the shift can be assumed to be damaged. Since biological materials and reagents are relatively scarce and tend to be expensive, no wasted material can be afforded. For this reason one needs to ensure that every single sample is completely processed by the end of the shift.

Obviously, a time *t* = *tshutdown* < *tshift* needs to be created after which no new raw material enters the automation line. Such shutdown time can be calculated by finding the time that the system throughput takes to fully enter the system given values for *AP* and *NP*.

$$tshutdown \approx TP \cdot \frac{AP}{NP} \tag{7.20}$$

After the system starts to shutdown, all stations one by one begin to turn off as the last part or batch of parts are processed. The time at which the last part enters Station *p* is named *tlast(p)*. As the number of parts coming into the system is *TP*, then to find when the last part enters Station *p* is the same as solving *Nin(p,t) = TP* where *t* is the unknown. *tlast(p)* may be approximated by multiplying the total number of parts *TP* by the effective arrival rate *Ape(p)* for Station *p*. The expression for *tlast(p)* is:

$$tlast(p) \approx TP \cdot Ape(p) \tag{7.21}$$

Likewise, there is a turn off time representing when the last part has been processed by Station *p toff(p)*. As the number of parts processed by the station is *TP*, *toff(p)* is calculated by solving *Nout(p,t) = TP* for t. *toff(p)* may be approximated by multiplying *TP* by the maximum number between the effective arrival period of Station *p* and the effective service time of Station *p* as when calculating *Nout(p,t)*.

$$toff(p) \approx TP \cdot \max\{Ape(p), Ste(p)\} \tag{7.22}$$

7.7. Queue Clearing Strategy

A common biological timing constraint is due to the fact that biological samples may get degraded if exposed to non-ideal environmental conditions for extended periods of time. In other words, the waiting time of samples lining at a queue of a given station must not exceed a maximum (which is presumably known) shelf-time limit denoted as *MQt(p).* The maximum allowable queue waiting time *MQt(p)* varies from station to station due to the varying biological properties of the products propagating in the system. Some products may be stable and thus may have very long shelf times. In such cases the station queue size limit may only depend on the physical space available for a queue *MQs(p)*. Other stations products may on the other hand be so unstable requiring essentially an immediate processing by the next station. Such stations must therefore have no queue at all (*MQt(p) = 0*). The timing constraint representing the maximum shelf time of each station can be transformed into an equivalent constraint involving a maximum queue size *MQs(p)* and vice versa. Recall the one to one relationship between *Qs(p,t)* and *Qt(p,t)* given by (7.16). Therefore:

$$MQs(p) = \left\lfloor \frac{MQt(p)}{St(p)} \right\rfloor \cdot Np(p) \tag{7.23}$$

Vise versa:

$$MQt(p) = \left\lceil \frac{MQs(p)}{Np(p)} \right\rceil \cdot St(p) \tag{7.24}$$

To find when the queues size (and its related queue time) for Station *p* reaches the maximum constraint value *MQs(p)* or *MQt(p)* a time *t* = *tpeak(p)* is defined. Hence, *Qs(p,tpeak(p))* = *MQS(p)* and *Qt(p,tpeak(p))* = *MQt(p)*.

The queue clearing strategy is based on system periodic input stopping and input resuming. Stopping (*tstop*) and resuming (*tresume*) times are pre-calculated to ensure that every queue for every station of the system adheres to its maximum queue restrictions (time and size equivalent limits). Between the stopping time and the resuming time no new parts enter the system as critical queues are present in the system and need to be cleared. At *t* = *tst*op samples are blocked from entering the system for a predetermined period of time *Tblock*. At *t* = *tresume*, samples start to enter the system again. Thus, *tresume* = *tstop* + *Tblock*. In general, the input to the system will have a period *tresume* with parts going into the system for an interval of time *tstop* and then no parts will enter for an interval of *Tblock*. In other words, *IN(t)* = *IN(t* + *n*·*tresume)* with samples entering the system for *n*·*tresume* ≤ *t* ≤ *n*·*tresume* + *tstop* and samples stopping for *n*·*tresume* + *tstop* < *t* < *(n* + *1)*·*tresume* for *n* = *0, 1, 2,….*

The stopping time is calculated as the time at which certain critical queues reach their allowed maximum limit. Therefore as a queue of Station *p* reaches its maximum allowed size *MQs(p)* it is assumed that at that time no more parts are coming from Station *p - 1*. For *t* ≥ *tstop* all queues in the system start to clear. At *t* = *tresume* the last remaining non-empty critical queue becomes empty and at that point in time the input to the system resumes.

In order to calculate the input stopping time *tstop*, four conceptual steps must be followed:

1. Calculation of the time *tpeak(p)* at which the queue waiting time or queue size reaches the value of *MQt(p)* or *MQs(p)* for all stations.

2. Identification of the critical station *CS*.
3. Calculation of the total number of samples that have entered the critical station prior to *tpeak(CS)*.
4. Calculation of the time that takes for the number of samples calculated in step 3 to enter the system. This will be *tstop*.

Step 1: Calculation of tpeak(p) for all stations p

As previously defined, *tpeak(p)* is the time at which Station *p* reaches the constraint condition *MQs(p)* or *MQt(p),* whichever is more restrictive. In order to identify which station has the tightest constraint and therefore will become the critical station *CS*, *tpeak(p)* needs to be calculated for every station. Once *tpeak(p)* for all stations are calculated, these values are compared. A shorter peak time means that its respective queue reaches the constraints values faster and therefore it would be a more limiting station.

The starting point to calculate *tpeak(p)* is expression (7.15). For reading convenience (7.15) is rewritten underneath:

$$Qs(p,t) \approx \left(\frac{t - tfirst(p)}{Ape(p)}\right) - \mathrm{mod}\left(\left(\frac{t - tfirst(p)}{Ape(p)}\right), Np(p-1)\right) + Np(p-1)$$
$$-\left(\left(\frac{t - ton(p)}{Ste(p)}\right) - \mathrm{mod}\left(\left(\frac{t - ton(p)}{Ste(p)}\right), Np(p)\right)\right) - Np(p)$$

Recalling some characteristics of (7.15), the modulus terms were included in the expression as *Nin(p,t)* and *Nput(p,t)* for any station are multiples of either *Np(p – 1)* or *Np(p)* respectively. Also as *Nin(p,t)* and *Nout(p,t)* have an error of ± *max{Np(p)}*, this error is passed on to (7.15). To manipulate the above expression, it is approximated by removing the modulus contribution of the expressions. By doing so the values for *Nin(p,t)* and *Nout(p,t)* are approximated to the trend line solution and the error will still be within acceptable margins. (7.5) is therefore approximated by the following expression:

$$Qs(p,t) \approx \left(\frac{t - tfirst(p)}{Ape(p)}\right) + Np(p-1) - \left(\frac{t - ton(p)}{Ste(p)}\right) - Np(p) \qquad (7.25)$$

First, the most limiting constraint for each station needs to be determined. For those stations that have both, *MQs(p)* and *MQt(p)* conditions, the most restrictive value is found by converting the *MQt(p)* value to a possible *MQs(p)* using (7.23) and comparing it to the given value for *MQs(p)*. Whichever value is smaller is the most restrictive case and will be used to calculate *tpeak(p)* for Station *p*. If only one condition is available for Station *p*, the available value is used. As *tpeak(p)* is calculated from (7.25) an expression for *Qs(p,t)*, for any *MQt(p)* constraint value or its equivalent *MQs(p)* needs to be calculated using (7.23).

Since *tpeak(p)* is defined as the time that the queue of Station *p* reaches the maximum allowable time, then *Qs(p,tpeak(p)) = MQS(p)*. Plugging in the values into (7.25) the expression becomes:

$$MQs(p) \approx \left(\frac{tpeak(p) - tfirst(p)}{Ape(p)} \right) + Np(p-1) - \left(\frac{tpeak(p) - ton(p)}{Ste(p)} \right) - Np(p) \qquad (7.26)$$

The only unknown variable in (7.26) is *tpeak(p)*. All other values are parameters of Station *p* or have been already calculated. Solving for *tpeak(p)* from (7.26):

$$tpeak(p) \approx \left(MQs(p) - Np(p-1) + Np(p) + \frac{tfirst(p)}{Ape(p)} - \frac{ton(p)}{Ste(p)} \right) \cdot \left(\frac{Ape(p) \cdot Ste(p)}{Ste(p) - Ape(p)} \right) \qquad (7.27)$$

Notice that (7.27) is only valid to calculate *tpeak(p)* when a queue actually exist. That is when *Ste(p) > Ape(p)*. If *Ste(p) ≤ Ape(p)* no queue form in Station *p* and the result of equation (7.27) is undefined or negative.

If time *tpeak(p)* for Station *p* is greater than *toff(p)* it means that the maximum queue size and therefore the maximum waiting time is not reached and thus, no clearing strategy needs to be implemented.

Step 2: Identification of a critical station CS

There is at least one critical station *CS* dictating where the most urgent tending of a queue is needed. This does not necessarily means that a queue clearing strategy is needed. As explained before, if *tpeak(p) > toff(p)* for all stations, existing queues are within the allowable constraints. If one or more queues violate the given maximum queue sizes and/or maximum allowed waiting time, such a station is the one selected in order to calculate the stop and resume

times for the queue clearing strategy. If the queue sizes and waiting times for the *CS* satisfy the timing and/or size constraints, every other station in the automation system will automatically meet the timing or length constraints.

CS is found by comparing all existing and valid *tpeak(p)* values for the stations that have queues. Station *p* that has the shortest *tpeak(p)* is considered the *CS,* therefore *CS* takes the value of *p* of the station with the shortest *tpeak(p)*. If Station *p* reaches first a conflicting queues size/time, it means that it has a tighter constraint and is therefore the most critical station. Critical station *CS* must satisfy:

$$tpeak(CS) < tpeak(p) \qquad \forall p \in [1,2,...,(CS-1)] \wedge [(CS+1),...P] \tag{7.28}$$

The station number *p* = *CS* is defined as the number of the Station *p* with the smallest *tpeak(p)*.

Step 3: Calculation of the number of samples that entered station CS prior to tpeak(CS)

The number of samples that entered *CS* prior to *tpeak(CS)* is calculated by finding the number of parts *Nin(p,t)* that have entered the given Station *p* by the time *tpeak(p)*. In other words, it is done by computing *Nin(CS,tpeak(CS))*. By (7.9):

$$\begin{aligned} Nin(CS, tpeak(CS)) \approx & \left(\frac{tpeak(CS) - tfirst(CS)}{Ape(CS)} \right) \\ & - \mathrm{mod}\left(\left(\frac{tpeak(CS) - tfirst(CS)}{Ape(CS)} \right), Np(CS-1) \right) + Np(CS-1) \end{aligned} \tag{7.29}$$

As *Nin(CS,tpeak(CS))* is the number of parts that entered the system up until when the limiting queue size/time is reached, if only these number of parts enter the critical station, no conflicting queues will be created. A larger value, beyond this value, causes the maximum queue size/time of the critical station to be violated. Need to keep in mind that the number of samples that enter the system are the number of samples that eventually enter the critical station. Therefore, if samples that enter the system are limited to *Nin(CS,tpeak(CS))*, the system will comply with the timing constraint *MQt(p)* and size constraint *MQs(p)*. As this is the maximum number of samples that can enter the system for all timing and queuing constraints to be satisfied it will be denoted as *MIN*.

$$MIN = Nin(CS, tpeak(CS)) \tag{7.30}$$

Step 4: Calculation of tstop

To compute the time at which samples should stop entering the system the maximum number of samples into the system, *MIN,* that has previously been calculated, needs to be taken into account. This value should prevent queues larger than the allowed by the constraint *MQt(p)* and/or *MQs(p)*. Samples only enter the system for as long as it takes the *MIN* parts to arrive. After *MIN* parts have entered the system, samples stop entering while queues clear.

The time at which samples stop entering the system *tstop*, is calculated by multiplying the number of maximum samples *MIN* by the effective arrival of the system *AP/NP*.

$$tstop = MIN \cdot \frac{AP}{NP} \tag{7.31}$$

As explained before, the queue clearing strategy involves three action times: *tstop*, to stop parts coming into the system; *Tblock*, for how long to stop the parts coming into the system; and *tresume,* the period the stopping and resuming operations have. And thereafter we repeat every *n·tresume* for *n* equal to zero or any positive integer value.

After computing *tstop*, the second parameter that needs to be calculated to implement the clearing strategy is *Tblock*: how long to wait before parts are allowed again into the system. In turn, this value represents the time that the critical queue takes to clear. The computation proceeds in a straightforward manner. *MQt(p)* is how long the critical queue takes to clear (as *Qt(CS,tpeak(CS)) = MQt(p)*). Thus:

$$Tblock = MQt(CS) \tag{7.32}$$

The system should resume inputting parts into the process line after the queue in the critical station has cleared. That is:

$$tresume \approx tstop + Tblock = MIN \cdot \frac{AP}{NP} + MQt(CS) \tag{7.33}$$

7.8. Critical Arrival Period

A second automation design scenario for handling long queues is to avoid the formation of all queues altogether. As explained before the only parameter that could be manipulated

without affecting the protocol is the effective arrival period to the global system *AP/NP*. The idea is to find a critical value for the global effective arrival period *AP/NP* in order to eliminate all queues. Below this critical value queues will be present. Above this value *TP* may start decreasing. This critical arrival period is denoted as *APc*.

Considering the slowest station *SS* which is known to have the longest effective service time, intuitively it can be said that in order to reduce the queue waiting time to zero (which is the same as saying that the queue size would be zero), the effective arrival period to the slowest station needs to be equal to the effective service time of the slowest station. By (7.25)

$$Qs(SS,t) \approx \left(\frac{t - tfirst(SS)}{Ape(SS)}\right) + Np(SS-1) - \left(\frac{t - ton(SS)}{Ste(SS)}\right) - Np(SS) \tag{7.34}$$

Need to solve the following equation for the unknown *Ape(SS)*:

$$0 = \left(\frac{t - tfirst(SS)}{Ape(SS)}\right) + Np(SS-1) - \left(\frac{t - ton(SS)}{Ste(SS)}\right) - Np(SS) \tag{7.35}$$

The solution of (7.35) is *Ape(SS)* = *Ste(SS)*.

By (7.3) (repeated below for reading convenience) we get that *APc* = *Ape(SS)* = *Ste(SS)*

$$Ape(1) = \frac{AP}{NP}$$

$$Ape(p) = \max\big(Ape(p-1), Ste(p-1)\big),\ p = 2,...P$$

$$Ape(p-1) = \max\big(Ape(p-2), Ste(p-1)\big).\ p = 3,...P \tag{7.36}$$

$$Ape(p) = \max\left(\frac{AP}{NP}, Ste(1), Ste(2),..., Ste(p-2), Ste(p-1)\right) \tag{7.37}$$

$$Ape(SS) = \max\left(\frac{AP}{NP}, Ste(1), Ste(2),..., Ste(SS-2), Ste(SS-1)\right) \tag{7.38}$$

Yielding the critical arrival period result.

7.9. Parallelizing Strategy

A third method to abide by queue times and sizes constraints is a parallelizing strategy by which some critical stations are duplicated. Implementing this method is of course more expensive but very often this is the only way to attain a desired throughput target. As already

explained, the requirements and constraints of a given protocol cannot be changed; the operations cannot be swapped around for the designers benefit; nor can the service time and timing constraints, for each of the stations, be altered.

The station parallelizing strategy might be implemented to meet one or two objectives:

The first parallelization objective, and the one fully explored in this dissertation, is to reduce queue sizes and waiting times to stay within allowable limits. Parallelizing might be a must under the scenario where a queue clearing strategy is not possible because *AP* might not be controllable due to storage equipment constraints. Sometimes indeed a stop and resume strategy of the systems input may not be possible.

A second parallelization objective would be to increase the automation system throughput. Such Throughput Enhancement strategy may be used even in cases where there is no problem with queues or products shelf time. In this chapter we shall assume that goal of the parallelizing process is to reduce queues sizes to meet all constraints. As it is done the throughput is increased as a pleasant by-product.

The idea behind the parallelizing strategy is to first identify stations deemed critical. Such stations become candidates for parallelization. Parallelization involves the duplication of the respective critical station. Initially two units are put in parallel with in the same critical station. The strategy involves a recursive algorithm by which after each duplication action the system is re-assessed. A station that has been duplicated, as it now can produce double as many parts, may or may no longer be considered critical. If it is still critical, a third parallel unit is added. If another station becomes more critical, a second unit is added to that station. The parallelizing process continues until all queue maximum waiting times are within the specified limits.

We conceptually view the parallelizing action as that of including new processing units to a given station in order to perform the same operation with more parts. These parts are processed simultaneously reducing the wait time for the samples residing within the station queue. As a new processing unit is introduced to a station the number of processing units per station is denoted as $Npu(p)$. An initial configuration with no parallel stations has $Npu(p) = 1$ for all p. As the parallelizing strategy is implemented, $Npu(p)$ for some stations p increases gradually

until the overall design requirements are met. As there is a finite floor space for the automation system, and also a finite equipment budget, a maximum number of additional processing units is often dictated by space and cost constraints. The final plausible solutions of this strategy are a collection of *Npu(p)* that describes which stations need to be parallelized and with how many units. The final vector solution for the parallelization strategy is the vector *NPU* containing the values of *Npu(p)* for all stations.

$$NPU = \left[Npu(1), Npu(2), ..., Npu(P-1), Npu(P)\right] \tag{7.39}$$

Parameters such as *tpeak(p)* and *CS* have already been defined for the queue clearing strategy and are equally valid to be used for the parallelization strategy as well.

The parallelizing strategy is an algorithm designed to find plausible system reconfiguration solutions. Unlike the Queue Clearing Strategy, the parallelizing strategy may not necessarily have a unique valid answer. Other than achieving the goal of reducing queue sizes and times, there are two algorithm terminating conditions that need to be used whenever the algorithm does not converge to a unique result or whenever system constraints are breeched. The first terminating condition is when the number of overall parallelized units exceeds a given maximum *MNPU* specified by equipment budget constraints. The second terminating condition is whenever the maximum number of parallelized units of a given station *MNpu(p)* is exceeded. Such a constraint is related to available floor space for such a station.

The procedure often features more than one plausible answer, and with other tools like the ones presented in the next chapter, an experienced designer may converge on the best implementation plan.

In order for all *MQs(p)* and *MQt(p)* to be met, the following iterative units parallelization strategy is proposed:

1. Calculation of the initial system *tpeak(p)* for all stations *p* to find if all queue sizes and queue times for the current system configuration are within the given constraints *MQs(p)* and *MQt(p)*. For the initial system *Npu(p) = 1* for all stations *p*.

- If the computed *tpeak(p)* > *toff(p)* for all stations, *Qs(p,t)* < *MQs(p)* and *Qt(p)* < *MQt(p)* for all stations *p*. Hence, the initial configuration *Npu(p)* = *1* for all *p*, is a plausible solution for the system and no further calculations are needed.

2. Identification of the critical station *CS* of the system and incrementing its *Npu(p)* by *1*.
3. Computation of the throughput *TP* for the new system configuration.
4. Comparing the number of parallelized units for each station against *MNpu(p)* to evaluate if floor space constraints are met.
    - If *Npu(p)* > *MNpu(p)* for any station, a floor space constraint is exceeded and the algorithm stops. The solution is not physically feasible.
5. Comparison of the total number of parallelized units in the system against *MNPU* to assess if budget constraints are not exceeded.
    - If ∑*Npu(p)* > *MNPU* the budget constraint is exceeded and the algorithm stops. The solution is not monetarily feasible.
6. Calculation of the current system *tpeak(p)* for all stations *p* to find if current queue sizes and queue times for the new configuration are within the given constraints *MQs(p)* and *MQt(p)*.
    - If the computed *tpeak(p)* > *toff(p)* for all stations, *Qs(p,t)* < *MQs(p)* and *Qt(p)* < *MQt(p)* for all stations *p*. Hence, the collection of current *Npu(p)* is a valid and plausible solution for the system and no further iterations are necessary.

Steps 2 to 6 repeat iteratively.

Let us next elaborate on each of the algorithm steps:

Step 1: Calculation of the initial system tpeak(p) for all stations p

The first step of the parallelization strategy procedure is to find if any of the stations *p* violates *MQs(p)* or *MQt(p)* constraints. For this purpose, (7.27) is used to compute *tpeak(p)* for all stations *p*. If *tpeak(p)* > *toff(p)* for all stations *p*, then queues for all stations are within the given

constraints for the given *tshift*. In this case, the initial system is a plausible solution and no further computations need to be performed. The initial configuration of *Npu(p) = 1* is a valid solution and is written as (7.39).

If this is not the case, some station or stations are violating the given constraints, and the procedure follows to *Step 2*.

Step 2: Identification of the critical station CS of the system

As for the queue clearing strategy, the critical station *CS* is determined by comparing the *tpeak(p)* for all stations. According to (7.28), Station *p* that has the smallest *tpeak(p)* is the most limiting station, thus becoming *CS*.

The identified *CS* is the first candidate station for parallelization. This means that an extra processing unit with the exact same parameters (*Np(p)* and *St(p)*) as the already existing unit (or units) in Station *CS* is added. The result of adding an extra processing unit is that Station *CS* will now be able to process *Np(CS)* parts more at the same time *St(CS)*. As now more parts are processed in the same service time, the effective service time for Station *CS* is altered, consequently the need to calculate the new *TP* as well as all other parameters for the new system configuration.

The new system configuration for the system is obtained by adding one extra unit to *Npu(CS)*, for all other stations *Npu(p)* stay the same.

$$\begin{aligned} &Npu(CS) \rightarrow Npu(CS)+1 \\ &Npu(p) = Npu(p) \qquad \text{For } \mathrm{p} \neq \mathrm{CS} \end{aligned} \tag{7.40}$$

It is a good practice to parallelize simultaneously a station that has the same service time as the critical station *St(CS)*. As at the moment of computation a downstream station with the same *St(p)* as the critical station is not an issue, it becomes problematic as the critical station is parallelized. Once parallelized, the former critical station will process more parts, and thus these parts could start to accumulate in the downstream station converting it in the future critical station.

Step 3: Computation of the throughput TP for the new system configuration

As previously described, due to the changes in the effective service time of the critical station, *TP* as well as other system parameters need to be recalculated. The new calculations need to reflect the inclusion of an extra process unit to the critical station.

A system parameter that cannot by modified under no circumstances is the actual service time for each of the processing units. That is, if samples need to be incubated for 10 minutes that is how long each of the samples needs to be incubated. Due to the addition of multiple processing units in some of the stations, extra *Np(p)* number of parts for each extra units can be processed at the same time. The previously defined effective service time *Ste(p)* in (7.2) needs to be extended to include the effect of introducing *Npu(p)* to the system. Using the same example as before, if an incubator station has a service time of *St(p)* = 10min, it means that a sample need to be incubated for 10 minutes. If a second unit is available, the station will be able to process two samples in those same 10 minutes. The actual service time for this station will be two samples every 10 minutes, or an effective service time of 5 minutes.

Keeping in mind the effect that *Np(p) > 1* has on *St(p)*, the inclusion of *Npu(p) > 1* has a similar effect: more than one sample are processed during the same *St(p)*. The number of samples that are processed in the same period of time become *Np(p)·Npu(p)*. The following expression for effective service time is extended to include both effects from *Np(p) > 1* and *Npu(p) > 1*.

$$Ste(p) = \frac{St(p)}{Np(p) \cdot Npu(p)} \tag{7.41}$$

The inclusion of *Npu(p) > 1* affects the complete system and it has to be reflected in the calculation of every system parameter. Therefore, in order to calculate *TP* using (7.19), *ton(p)* and *OUT(t)* need to be extended to incorporate the impact of *Npu(p) > 1*, and the values for the new system configuration for *Ste(p)*, *Ape(p)*, *tfirst(p)*, and *ton(p)* need to be recalculated.

The impact of *Npu(p) > 1* is similar: the station processes *Np(p)·Npu(p)* number of parts at the same time. To extend *ton(p)* and *OUT(t)* this result is used.

In conclusion, in order to be able to calculate *TP* for the new configuration the following steps are taken:

1. The new values for effective service times *Ste(p)* are calculated using (7.41).
2. The new values for effective arrival periods *Ape(p)* are calculated using (7.3).

$$Ape(1) = \frac{AP}{NP}$$
$$Ape(p) = \max\left(Ape(p-1), Ste(p-1)\right), p = 2,...P$$

3. The values for *tfirst(p)* are recalculated using (7.5).

$$tfirst(1) = 0$$
$$tfirst(p) = ton(p-1) + St(p-1), p = 2,..., P$$

4. The expression for *ton(p)* is extended to include the effect of *Npu > 1* and is used to compute the new values.

$$ton(1) = tfirst(1) + \left(\left\lceil \frac{Np(1) \cdot Npu(1)}{NP} \right\rceil - 1\right) \cdot AP$$
$$ton(p) = tfirst(p) + \left(\left\lceil \frac{Np(p) \cdot Npu(p)}{Np(p-1) \cdot Npu(p-1)} \right\rceil - 1\right) \cdot \max\left(St(p-1), Np(p-1) \cdot Npu(p-1) \cdot St(p-2)\right) \tag{7.42}$$

5. The expression for *OUT(t)* is extended.

$$OUT(t) = \left(\frac{t - ton(P)}{\max\{Ape(P), Ste(P)\}}\right) - \mathrm{mod}\left(\left(\frac{t - ton(P)}{\max\{Ape(P), Ste(P)\}}\right), Np(P) \cdot Npu(P)\right) \tag{7.43}$$

6. Finally, *TP* for the new configuration can be calculated using (7.19).

$$TP = OUT(tshift)$$

Step 4: Comparing the number of parallelized units for each station against MNpu(p)

The floor space constraint is added by limiting the number of units a station can have. The space available for multiple processing units could be translated into a maximum number of process units that can fit into a given space. Thus we use the maximum number of units for each station *MNpu(p)* to evaluate if the given solution could fit into the available space. If one of the *MNpu(p)* constraint is violated, the solution will not be physically realistic and the configuration must be discarded.

$$\text{if} \quad Npu(p) > MNpu(p) \ \text{ for any p} \Rightarrow NoSolution \tag{7.44}$$

Step 5: Comparison of the total number of parallelized units in the system against MNPU

Likewise, the budget constraint is added by limiting the maximum number of processing units in the entire system. The budget can be translated to a maximum total number of equipment present in the production line. If the maximum number of units is exceeded, the solution will be very costly and the available budget will not suffice.

$$\text{if} \quad \sum Npu(p) > MN\text{PU} \Rightarrow NoSolution \tag{7.45}$$

Step 6: Calculation of the current system tpeak(p) for all stations p

Finally, the new system configuration *tpeak(p)* values need to be recalculated to determine if the current queue sizes and queue times for the new configuration are within the given constraints *MQs(p)* and *MQt(p)*.

Again, as for the new configuration at least one station has a *Npu(p) > 1*, (7.27) needs to be extended to include multiple processing units effect. Thus the expression for *tpeak(p)* for *Npu(p) > 1* becomes:

$$tpeak(p) \approx \left(MQs(p) - Np(p-1) \cdot Npu(p-1) + Np(p) \cdot Npu(p) + \frac{tfirst(p)}{Ape(p)} - \frac{ton(p)}{Ste(p)} \right) \cdot \left(\frac{Ape(p) \cdot Ste(p)}{Ste(p) - Ape(p)} \right) \tag{7.46}$$

In order to be able to evaluate if queue sizes and queue times are within the given constraints *MQs(p)* and *MQt(p)*, *toff(p)* needs to be recalculated for the new configuration using (7.22).

$$\begin{aligned} &\text{if} \quad tpeak(p) > toff(p) \ \text{ for all p} \\ &\Rightarrow (Qs(p,t) < MQs(p)) \wedge (Qt(p,t) < MQt(p)) \\ &\Rightarrow ValidSolution \end{aligned} \tag{7.47}$$

This sequence of five steps continues until one of the three termination conditions occur: *Npu(p) > Mnpu(p)* for at least one station *p*, ∑*Npu(p) > MNPU*, or *tpeak(p) > toff(p)* for all stations *p*.

7.10. Parallelizing Strategy Numerical Example

Going back to the previous chapter and the 17 stations case study used throughout the dissertation, the system is used once again in this chapter to evaluate the validity of the Parallelizing Strategy previously described.

Table 7.4 shows the results for the implementation of the Parallelization Strategy to the 17 stations system. Above the table, the service time for each of the station is listed. In this case, *Np(p) = 1* for all stations. The first column of the table shows the *AP* and *MQt* parameters used for the example and the *TP* obtained for the configuration. The second column shows the number of parallel units, *Npu*, used for the configuration. And the following 17 columns shows the values for the maximum value for queue size and queue time that each of the stations had at any given time within a period *tshift*. There are two examples in the table. The first four rows comprise the iterations for Example 1, and the last 6 rows comprise the iterations for Example 2.

Example 1: *AP = 30* and *MQt(p) = 3600s* for all stations *p*.

The first row of the example shows the values for the different queue sizes and queue times of all stations in the system before parallelization. This will correspond to step 1 of the described methodology. For this first configuration, the *TP = 470*, and Station *5* is the conflictive station as its *Qt(p,t)* during the shift time accumulates up to *14040s*. For this Station *5* *Qt(p,t) > MQt(p)* thus, the first candidate configuration will include *Npu(5) = 2*.

The second row of the table shows the results for the second configuration. Notice that *St(5) = 120s* and that Station *14* and *16* have the same service time. As this Station *5* was parallelized in the first iteration, Station *14* with the same service time as Station *5* is now the problematic one. Since they have the same effective service time, parallelizing only Station *5* does not increase *TP*.

Table 7.4 Numerical results for the Parallelizing Strategy

St(p)			10	10	20	10	120	15	10	15	10	15	10	15	10	120	10	120	10
Station			S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17
AP = 30 MQt(p)= 3600 TP = 470	No parallel units	MQs	0	0	0	0	235	1	0	0	1	0	1	0	1	1	1	1	1
		MQt	0	0	0	0	14040	15	0	0	10	0	10	0	10	10	10	10	10
	Station 5 Npu = 2	MQs	0	0	0	0	3	3	0	0	1	0	1	0	1	235	1	1	1
		MQt	0	0	0	0	130	45	0	0	10	0	10	0	10	14090	10	10	10
	Station 14 Npu = 2	MQs	0	0	0	0	3	3	0	0	1	0	1	0	1	3	3	235	1
		MQt	0	0	0	0	130	45	0	0	10	0	10	0	10	65	30	14120	10
AP = 30 MQt(p) = 3600 TP = 920	Stations 5, 14, 16 Npu = 2	MQs	0	0	0	0	3	3	0	0	1	0	1	0	1	3	3	3	3
		MQt	0	0	0	0	130	45	0	0	10	0	10	0	10	65	30	50	30
AP = 10 MQt(p) = 3600 TP = 470	No parallel units	MQs	0	0	235	0	313	1	0	0	1	0	1	0	1	1	1	1	1
		MQt	0	0	4685	0	18720	15	0	0	10	0	10	0	10	10	10	10	10
	Stations 5, 14, 16 Npu = 2	MQs	0	0	235	0	158	3	0	0	1	0	1	0	1	3	3	3	3
		MQt	0	0	4685	0	4900	45	0	0	10	0	10	0	10	65	30	50	30
	Station 5 Npu = 3	MQs	0	0	235	0	5	5	0	0	1	0	1	0	1	158	3	3	3
		MQt	0	0	4685	0	240	75	0	0	10	0	10	0	10	4870	30	50	30
	Station 14, 16 Npu = 3	MQs	0	0	235	0	5	5	0	0	1	0	1	0	1	5	5	5	5
		MQt	0	0	4685	0	240	75	0	0	10	0	10	0	10	175	50	130	50
	Station 3 Npu = 2	MQs	0	0	1	1	236	5	0	0	1	0	1	0	1	5	5	5	5
		MQt	0	0	10	10	5120	75	0	0	10	0	10	0	10	175	50	130	50
AP = 10 MQt(p) = 3600 TP = 605	Station 3 Npu = 2 Station 5, 14, 16 Npu = 4	MQs	0	0	1	1	205	7	0	0	1	1	1	1	1	7	7	7	7
		MQt	0	0	10	10	3585	105	0	0	10	5	10	5	10	345	70	250	70

The purpose of Example 1 is to illustrate how the strategy works, but also to explain why it was recommended in the second step of the procedure, to parallelize at the same time downstream stations that have the same service time as the critical station. The results of the second iteration are still not valid as Station *16* has not been parallelized. Only until Station *5*, *14*, and *16* have been parallelized, the effective service time, and thus, for the effective arrival time decreases and *TP* increases. After the third iteration a valid configuration is obtained. Using (7. 39), the valid configuration for the 17 station system is (forth row in the example):

NPU = [1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1]

TP = 920

At the end, the configuration resulting from the parallelization strategy had 3 processing units more than the initial system, which represented $12,900 increase in cost from an initial of $120,000, and an increase in *TP* of almost 50%.

Example 1 shows the complete progression of the algorithm starting with no parallel stations until a feasible solution that meet the queuing constraints was found.

Example 2: *AP = 10* and *MQt(p) = 3600* for all stations *p*

This second example works in the same way as the first one. Starting with the values for the initial configuration of the system, each following row is an iteration of the previous configuration. In Example 2, five different iterations and thus system configurations were evaluated. The valid result is

NPU = [1, 1, 2, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 4, 1, 4, 1]

TP = 605

In Example 2 some interesting conclusions can be drawn. For the calculations *MNPU* and *MNpu(p)* were kept very relax to let the strategy run until a valid solution was found. The resulting configuration for Example 2 had 14 processing units more than the original system. This means that it almost doubled the number of processing units available in the system. That translates directly into a $ 54,000 more than the initial system which costs $120,000, and a *TP* increase of 30%.

From comparing both results, it is clearly demonstrated that reduction of *AP* does not increase *TP* if *AP* < *Ste(SS)*. One can intuitively say that the configuration resulting from the first example is the most reasonable one. But if a limiting requirement exist for *AP* = *10s*, then the only possible solution will be the second one.

The examples presented for the parallelization strategy introduces the need for tools to be able to compare plausible configuration solutions and decisively chose the better one with founded criteria. Chapter 8 formulates a system configuration selection criterion that includes, automation line cost, throughput, and floor space to solve this problem.

8. DESIGN METHODOLOGY FOR AUTOMATION SUB-SYSTEMS CONFIGURATION SELECTION

8.1. Introduction

Design of an automation line is a multi-objective optimization problem involving throughput, yield, floor space and cost constraints. This chapter examines the feasibility of computer-aided automation design for biotechnology applications using Arena® software. Automation design necessarily involves a tradeoff among conflicting objectives. Ultimately all objectives (throughput, yield, space utilization and process cost) may be cast in a unified cost structure to allow an optimal design decision. This chapter demonstrates such a design methodology.

In Chapter 2 the importance of and need for automated solutions for biotechnology processes is illustrated by means of a specific process for preparation of RNA from tissue-cultured cells. Constraints that biotechnology equipment typically has are highlighted in Chapter 6. With no station-by-station yield control biotechnology operations have to be very accurate. It is conceptually feasible to monitor product quality of each station, not allowing defected samples to proceed to the next station; however it is beyond the scope of this dissertation. In order to assure in open loop that the output of each station is within specifications, the operations require precision. Another feature of many biotechnology processes is the relative high cost of processed materials. The samples and reagents involved in such processes tend to be expensive and sometimes very scarce and therefore waste must be kept to a minimum. Many biotechnology process steps may require stringent environmental conditions and tight timings constraints.

As a manual biotechnology protocol is extrapolated to the automation domain and because of the vast variety of equipment available there are often numerous different possible layouts of the production line. Some equipment may be shared, as explained in Chapter 2. Some equipment may be duplicated as explained in Chapter 7. The purpose of the design processes

presented in this chapter is to be able to select the best solution among finitely many candidate configurations that are evaluated for realizing the same protocol. To accomplish such a configuration selection task, for any protocol that requires automation, one has to find first any working configuration solution; second, such a plausible configuration is to be simulated by one of the commercial Industrial Engineering software tools such as Arena® in order to establish a baseline throughput performance; third, more plausible automation configurations are identified; finally, by applying to each candidate configuration an appropriate selection criterion, the best solution is chosen.

The variety of Automation Modules discussed in Chapter 6 serves as a source for finding plausible working configuration solutions. Such Automation Modules, along with the guidelines and methods described in Chapter 6 are employed to create the models for the biotechnology protocols for subsequent simulations. Using the Automation Modules helps to assess the equipment that is needed, the transportation method needed between pieces of equipment, and it eventually provides a comprehensive set of feasible configurations.

For each candidate configuration, the systems have to be evaluated from timing and queuing perspectives to assure the meeting of all the protocol constraints. In order to carry out the timing and queuing study, the methodology and equations defined in Chapter 7 are assumed to be used as needed.

For comparison purposes each of the automation configuration candidates needs to be assessed quantitatively: cost, footprint and throughput. This may be done manually or by simulation. In particular, the throughput of each configuration may be calculated by the approximated throughput formula developed in Chapter 7, or may be approximately determined by the MATLAB® Simulator developed in Chapter 7, or it may be calculated exactly using a simulator such as Arena®. The MATLAB® simulator may be used to assess configurations that do not involve equipment sharing. The reason for that is that the Simulator does not include product labeling steps and does not keep track of every single sample. In addition, as transportation time between stations was assumed throughout Chapter 7 to be negligible it adds error to the respective throughput calculations

To illustrate such design configuration selection process by means of a simple example, we first extract a sequence of a few steps from the full preparation of RNA from tissue-cultured cells protocol. The validity of the configuration selection process is demonstrated for both batch productions as well as for high-throughput productions. Thereafter, limited number of configurations proposed for implementing the complete case study described in Chapter 2 are discussed.

The configuration selection method proposed in this chapter is based on a quantitative Selection Ratio criterion for assessing alternative automation sub-system configurations. The criterion involves the process cost, space utilization and throughput (Cardona, 2009, 2012a).

8.2. The Economic Basis for Automation Configuration Selection

The candidate automation configurations are evaluated through a comparison of cost, footprint, and throughput.

The capital costs of an automation line construction are the one-time fixed construction expenses that are necessary. Such expenses may include the purchase of land or the cost of space acquisition by rental, the cost of design and construction of the line, and equipment purchases (Investopedia, 2014). Some of these capital expenditures depend on the specific configuration selected for the automation line and some are independent of such configurations. The costs incurred whenever purchasing hardware and software for the automation line is part of the production line cost. Another part of the cost relates to the cumulative operational cost of the production line. For example, the total space rental cost for the lifetime of the automation line, or likewise the total utilities cost. Importantly, the cost of space acquisition depends on the total area (footprint) needed for the implementation and as such it too is dependent of the configuration selected.

Denote by C_{PL} that part of the total production line cost that is dependent upon the automation line configuration selection.

$$C_{PL} = C_E + C_{FP} \tag{8.1}$$

Where C_E is the total cost of equipment and C_{FP} is the total cost for space acquisition. The *FP* subscript stands for “footprint”. The cost C_E depends on the number of equipment pieces

that need to be purchased and cost of each unit whereas C_{FP} depends on the total footprint area *FP*. Both cost factors strongly depend on the particular choice of automation line configuration.

Operating costs of the production line are divided into two broad categories: fixed operating costs (FOC) and variable operating costs (VOC). FOC includes salaries, rent, and utility bills. Since FOC has to be paid regardless of the operation configurations of the system, these are completely independent of the layout of the production line, and therefore not included for the evaluation of the candidate configurations. VOC includes costs associated with the disposable labware and consumable materials (both the input samples as well as the solutions and reagents) depending on how many samples are processed. The VOC costs clearly depend on the automation line throughput and the throughput of course depends on the configuration selection.

Let us denote the daily VOC as C_V where

$$C_V = C_{VU} \cdot N_U \tag{8.2}$$

Where C_{VU} is the variable cost per unit, and N_U is the number of units produced per day.

The daily operating cost C_O consists of both a fixed and a variable term:

$$C_O = C_F + C_V = C_F + C_{VU} \cdot N_U \tag{8.3}$$

Where C_F represents the daily fixed operating cost.

Assigning some arbitrary price per unit *P* to the production line product, the daily income is:

$$I = P \cdot N_U \tag{8.4}$$

The daily profit *R* is therefore:

$$R = I - C_O = P \cdot N_U - C_F - C_{VU} \cdot N_U \tag{8.5}$$

It is assumed that *P* is selected such that *R* is positive. There are no other restrictions on the choice of *P*.

The configuration-dependent daily profit R_C is therefore defined as:

$$R_C = (P - C_{VU}) \cdot N_U \tag{8.6}$$

Note that both *P* and C_{VU} are independent of the automation configuration choice.

The value of the configuration-dependent daily profit divides the total cost of the production line to yield T_{total} the number of days to return the total investment and such a figure is too configuration-dependent. Let $T < T_{total}$ denote the total time (in days) for Return on Investment (ROI) for C_{PL} (the configuration-dependent portion of the automation line capital cost) based on the daily profit R_C.

$$T = \frac{C_{PL}}{R_C} = \frac{C_E + C_{FP}}{(P - C_{UV})N_U} = \frac{C_{PU}}{P - C_{UV}} \tag{8.7}$$

Where C_{PU} the production line cost per unit product is defined as:

$$C_{PU} = \frac{C_{PL}}{N_U} \tag{8.8}$$

Because the configuration-dependent production line cost C_{PL} is independent of both the product unit price P and the operating variable cost C_{UV} it is seen in (8.7) that T is proportional to C_{PU}. A configuration selection that makes C_{PU} as small as possible will make T the smallest.

In (8.1) - (8.7) only equipment, labware, and consumable material that depends on a specific configuration and its throughput have been considered. In the classical ROI analysis value many other cost items such as: the cost of designing the line, the cost of building the facility, the monthly utilities, and transportation cost among many others are all taken into consideration. The true time that it takes to return the investment is obviously many times longer than T that was calculated above.

The reciprocal of the cost per unit, C_{PU}, is thence chosen as a Selection Ratio, SR for the candidate automation line configurations. Whenever comparing different automation implementation configurations, the larger the ratio between the daily throughput and the cost of the production line the better is the configuration.

$$SR = \frac{N_U}{C_{PL}} \tag{8.9}$$

There are some necessary conditions that if not met, the configuration must be discarded and not be compared with the others:

1. The production line cost should be within a given budget

2. The footprint of the system should be within the space available
3. The throughput of the system should meet the requirements of the protocol

The configurations that meet these three conditions are thereafter compared by examining the values for *SR*:

1. The candidate configuration should have the highest *SR* value.
2. If two configurations have exactly the same *SR* value, then the system with the higher throughput would be the best solution.

A somewhat surprising deduced consequence is that the choice of the best configuration is independent of the price set to the product. Finding the best configuration depends only on the cost of the equipment used and the throughput provided by the configuration.

8.3. Example 1: Five-Station-Process

The sequence of steps of the RNA purification automation process, chosen to illustrate the way the configuration selection criterion works, involves repetitive centrifuging operations to perform separation of biochemical substances.

This subset of operations chosen for the computer-aided design optimization example represents a set of operations that is present in the isolation of RNA as well as in many other biotechnology protocols. These are Stations 14 through 17 in the automation line discussed in Chapters 2 and 5 that as we recall require pick-and-place manipulation. This group of process steps is also interesting from an optimization point of view due to the consecutive centrifuging steps that may involve multiple centrifuges as well as multiple robots, or some measure of equipment sharing. This same subset of operations is included in other biotechnology protocols whenever washing or eluting of a sample is needed, or simply if more than one reagent is added and a subsequent separation operation is thereafter needed.

The set of operations consists of five stations: the first station is where the sample arrives into the system. It could be coming from a storage place, or from a previous step in the protocol. The second station is a centrifuge station where the sample is centrifuged for a certain amount of time (two minutes in this example). The third station performs dispensing. In this case study a wash buffer is added to the sample. The fourth station is a second centrifuge station. Finally, the

fifth station is where the sample exits the system, possibly to the next station in a longer process or, like in this case, to a storage unit because the output of this station happens to be the overall final product. A block diagram of the subset of operations is shown in Figure 8.1.

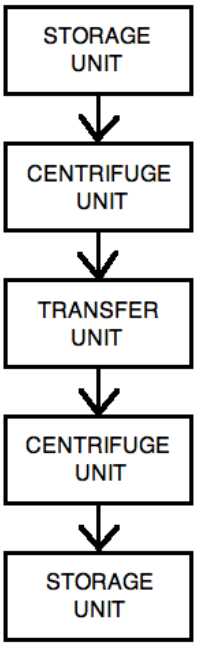

Figure 8.1 Subset of operations studied in the example

8.3.1. Candidate Configurations for Automation Design

Four possible configurations for batch production of RNA are explored. In order to have a consistent optimization study, the same equipment and labware is employed in all configurations simulations. This ensures that the conclusions drawn are due to the difference in automation configuration and not due to the difference in equipment and/or labware. The idea is to evaluate these four possible configurations comparing throughput, floor space and cost: the cost of producing the samples and the value that these samples represent. A set-up that has the best return on investment for a given product pricing strategy might be considered the best implementation option.

The batch production automation layout candidates for this example all represent “low-throughput” solutions, whereas high-throughput solutions are explored in Section 8.4. It is important to keep in mind that any high-throughput automation strategy necessitates some

customized equipment. The investment in customized tooling makes sense if the actual increase in potential revenue is significant due to the considerable increase in throughput.

Referring to Chapter 6, all the instruments involved in this five-step-process can be integrated by means of Automation Modules. The equipment used - centrifuges, dispensers, and a storage unit - follow the same operational principle: each piece of equipment can carry out a single operation, and thus can be integrated into an automation line by means of Automation Module AM1.

8.3.1.1. Configuration 1: Dual robots and dual centrifuges arranged linearly

The first configuration involves the assignment of a robot for each centrifuge. Each of the robots loads/unloads just one centrifuge as it transfers the labware to and from the centrifuge station. Below is a detailed description of each of the stations for Configuration 1:

Station 1: Represents the previous station where the sample enters the system. In this specific case and for the purpose of modeling and simulation the sample is assumed to be in storage prior to entering the system.

Station 2: In the Centrifuge Station the sample is centrifuged for 2 minutes. In the simulations a standard centrifuge is used that has a rotor that can fit up to six 1.5ml collecting tubes.

Station 3: The Dispensing Station consists of a commercial dispensing unit where a reagent is added to the column and to the collecting tube.

Station 4: A second centrifuge station is identical to Station 2.

Station 5: A storage unit is where the sample exits the system. The samples are kept in a storage unit for future use.

Since centrifuges can only be loaded and unloaded from the top a pick-and-place device is needed to perform such an operation. The best solution is to use a robotic arm capable of loading and unloading the equipment, and help with the manipulation of labware among stations. The layout for Configuration 1 is shown in Figure 8.2.

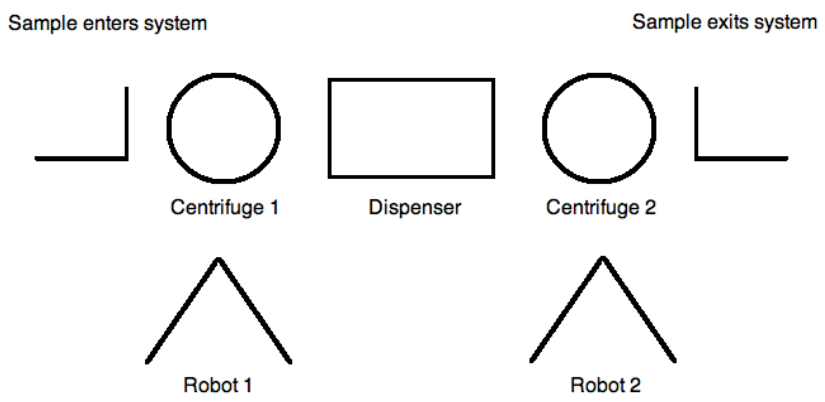

Figure 8.2 Conceptual diagram for Configuration 1

As this configuration does not share any equipment, it can be simulated with the MATLAB® simulator developed in Chapter 7 to provide guidance as for the maximum throughput possible.

8.3.1.2. Configuration 2: Single robot serving two centrifuges in a circular automation line geometry

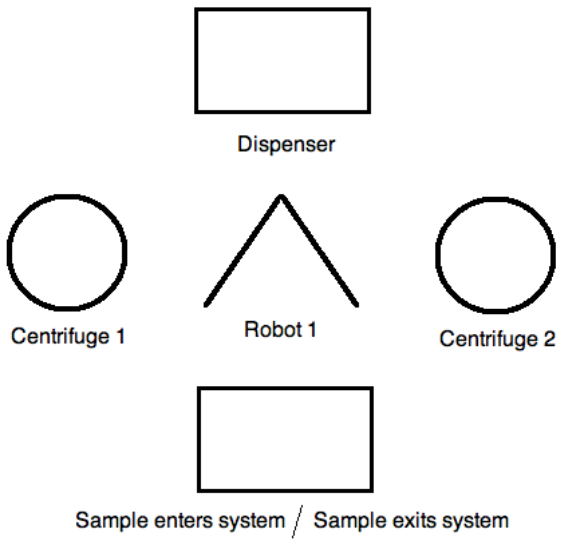

Figure 8.3 Conceptual diagram for Configuration 2

Robots are relatively among the more expensive pieces of hardware in the automation line, if not the most expensive. The second configuration is a setup aimed at cost reduction. The two centrifuge stations share one robot that loads and unloads tubes to and from both centrifuges. A circular geometry enables all stations to lie within the reachable workspace of the

robot. The circular geometry also enables the sharing of storage units, Configuration 2 layout diagram is shown in Figure 8.3.

From this point and on, the recommendation of redundant infrastructure from the integration methodology described in Chapter 6 is taken into consideration. Candidate configurations, as are Configurations 2, 3, and 4, involving equipment sharing are evaluated as well.

8.3.1.3. Configuration 3: Dual robots and dual centrifuges arranged in a circular geometry

Configuration 3 conceptually retains the throughput obtained whenever a centrifuge utilizes a dedicated robot for each centrifuge. The only caveat is to prevent the robots from having intersecting workspaces. Configuration 3 layout diagram is shown in Figure 8.4. Configuration 3 has the same stations and structure as Configuration 2.

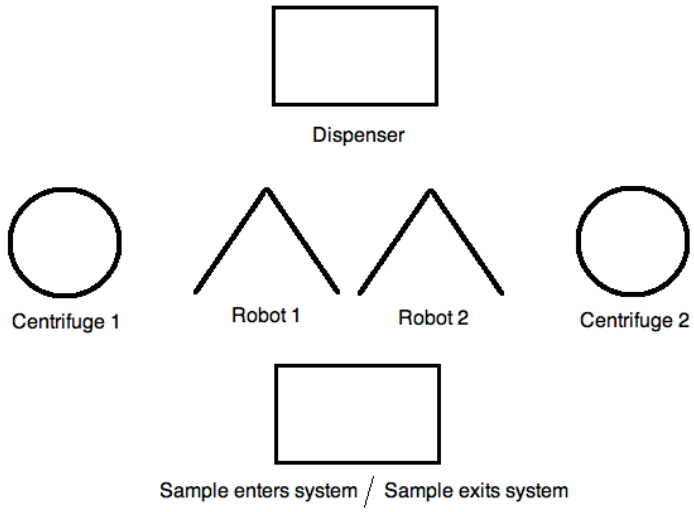

Figure 8.4 Conceptual diagram for Configuration 3

8.3.1.4. Configuration 4: Single Robot and single centrifuge strategy in a circular geometry

The fourth configuration aims at reducing both footprint and equipment cost, paying by having a lower throughput. Reachable robot workspace constraints dictate a circular geometry. The same storage unit is used to store “samples” – initial product, and RNA the “final product”. The layout diagram of Configuration 4 is shown in Figure 8.5. The single centrifuge station performs both centrifuging process steps.

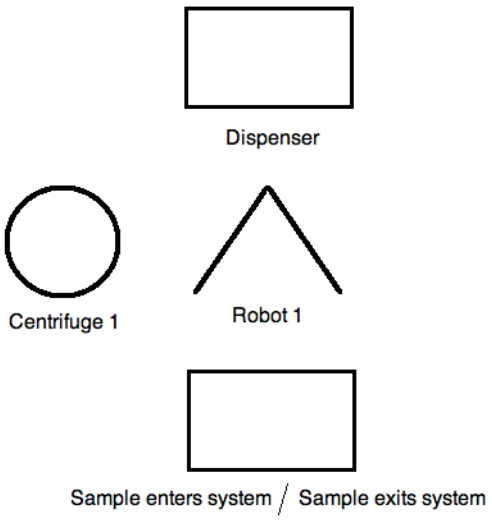

Figure 8.5 Conceptual diagram for Configuration 4

8.3.2. Configurations Simulations with Arena®

The four candidate configurations (discussed in Section 8.3.1) were modeled in Arena® to obtain the throughput information. The total footprint information was calculated based primarily on the robots workspaces.

The Arena® block diagram for the first three configurations is shown in Figure 8.6. The difference between the first three evaluated solutions has to do with the way labware is handled and manipulated. Arena® uses tables of resources and parameters to determine the transportation of the labware between samples. In these tables, the different transporters are fully described by their distance, size, and stations service parameters. Configuration 1 and 3 have each two transporters and the tables fully describe which station each transporter has to serve, the distance between stations, and the priority of the serving stations. Configuration 2 only has one transporter and the table describes the distance between stations and the priority to service the different stations.

The Arena® model used to simulate Configuration 4 (Figure 8.7) is considerably different than the ones discussed above. In this case a decision table needs to be included in order for the transporter to take the sample to the right station after it has been served. In the simulations it was assumed that the line runs a single eight hours shift a day. The entities (modeling the tubes containing the sample material) were assumed to be entering the system every 60 seconds.

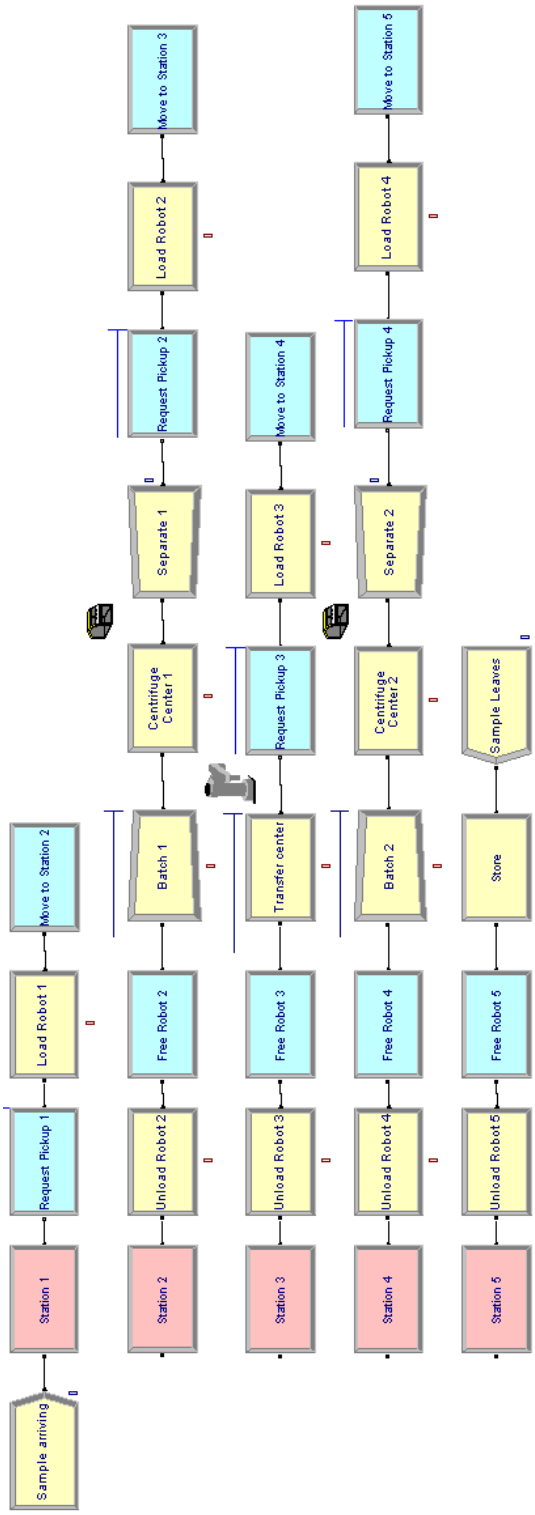

Figure 8.6 Arena® model for Configuration 1, Configuration 2 and Configuration3

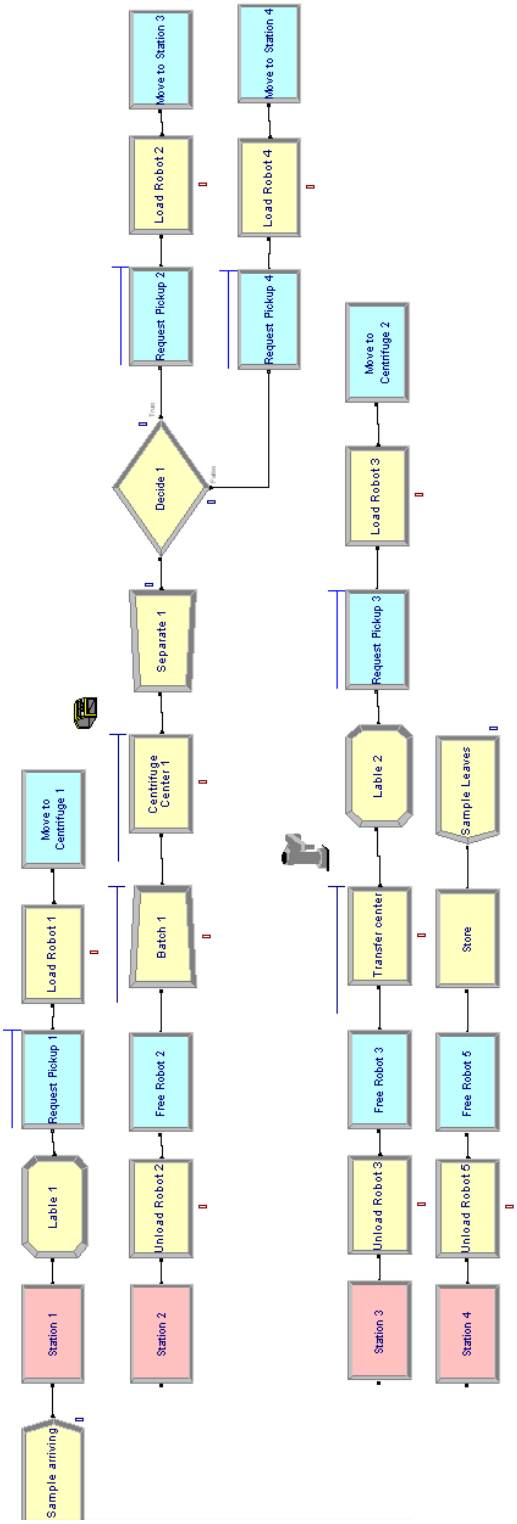

Figure 8.7 Arena® model for Configuration 4

8.3.3. Cost Figures for Equipment Used for the Numerical Example

In order to allow quantitative assessment we consider typical present day costs of commercially available equipment (obtained from web resources or by direct price quotes). These are listed below:

Robot (*certified for biological applications*):

- Fanuc M-430iA (FANUC, 2014) (Figure 8.8)
- Price: $54,900
- Footprint: 320 x 320 mm

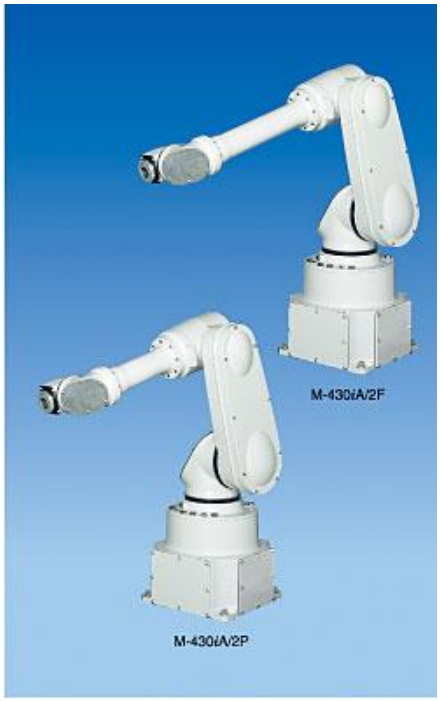

Figure 8.8 Fanuc M-430iA (FANUC, 2014)

Centrifuge:

- Kizker MZ011
- Price: $3933
- Footprint: 275 x 347 mm

Rotor:

- Kizker MZ011_2424
- Price: $305

Figure 8.9 Centrifuge 5427R (Eppendorf, 2013)

Standard Equipment:

- Dispenser: 600 x 400 mm
- Storage Unit: 1100 x 600 mm

Labware:

Columns & Reagent:

- QIAGEN – Rneasy Mini Kit (QIAGEN, 2006b) (Figure 8.10)
- Price: $227.00 per 50 units
- Centrifuge tubes:
- Kizker G013 True-Lock 1.5ml Tubes (Figure 8.10)
- Price: $36.50 per 1000 units (QIAGEN, 2006b)

Figure 8.10 Centrifuge tubes and columns (QIAGEN, 2006b)

8.3.4. Numerical Example

Table 8.1 provides simulations throughput results for an 8-hour shift. Throughput is defined as the total number of units prepared per single shift. Time per sample is the average time to produce one sample.

Table 8.1 Simulation throughput results for 5-Station protocol

Configuration	Throughput (number of samples)	Time per sample
1	645	45 sec
2	523	55 sec
3	645	45 sec
4	534	54 sec

If the only objective of automation is to maximize throughput, the configurations that use two robotic arms, one to handle each centrifuge station, clearly have a greater throughput. In one 8-hour shift a two-robots solution produces 122 samples more than a single robot implementation. Another way to quantify the optimality of the two-robot solution is to say that each sample is produced 10 seconds faster, compared with the single robot implementation. We note that throughput for two-robot solutions is not twice larger than for one-robot solutions. What dominates the throughput computation in this numerical example is the time that the centrifuges need to run which is independent of the configuration selection. The queues to each centrifuge do vary with the configurations but their impact on the total throughput is secondary. The cost of the product line in this example is dominated by the unit cost of a robot arm. Doubling the number of robots almost doubles the production line cost; however the throughput does not increase by the same rate.

Using (8.1) – (8.5) the cost of the production lines, income, revenue, and return of investment can all be calculated.

Table 8.2 Price of equipment used

Equipment	Price ($)
Robot	54,600
Centrifuge + Rotor	4,238
Dispenser	2,500
Storage Unit	1,200

Typical equipment prices are found in Table 8.2. Since Arena® cannot automatically draw the layout of the equipment; the computation of the floor space has to be done manually. In order to calculate the footprint the configurations are sketched, and the total square footage needed to setup the equipment is found. The total area is then multiplied by 1.5 to have space to accommodate other equipment not included in the model like storage for the labware, and additional space for the operators to work. For the sake of example the capital cost per square foot is assumed to be $40.00. An alternate way to include the floor space cost is to add it to the line operating cost (e.g. $0.5 per square foot per one day shift for commercial rented space). Whenever laying out the equipment, the first consideration to be made is that the robot should be able to reach the different units that it serves. The robot used in this example can reach any point within 90cm from its base center. The equipment modules were spaced 20cm apart. As a method of providing safety margin as well as a simple way of estimating the production line area, a rectangle containing the robots circular reachable space is created and this is called the Peripheral Margin. The total area is then multiplied by a factor of 1.5 as explained earlier. Figures 8.11 to 8.14 show the layout and total area for the different configurations.

Table 8.3 Layout results

Configuration	Peripheral Margin (cm)	Area (sq-ft)	FP value (sq-ft)
1	1080	69.75	104.63
2	720	34.87	52.30
3	760	38.75	58.12
4	720	24.87	52.30

The resulting peripheral margin, area and footprint (*FP*) value for each configuration appear in Table 8.3. In this example the cost associated with floor space plays a negligible role. This may be the situation for most biotechnology applications. In other words, tight floor space constraints may be used just to screen out bad candidate automation configurations, but play only a minor role in the selection process of best configurations among those that do meet the floor space constraint.

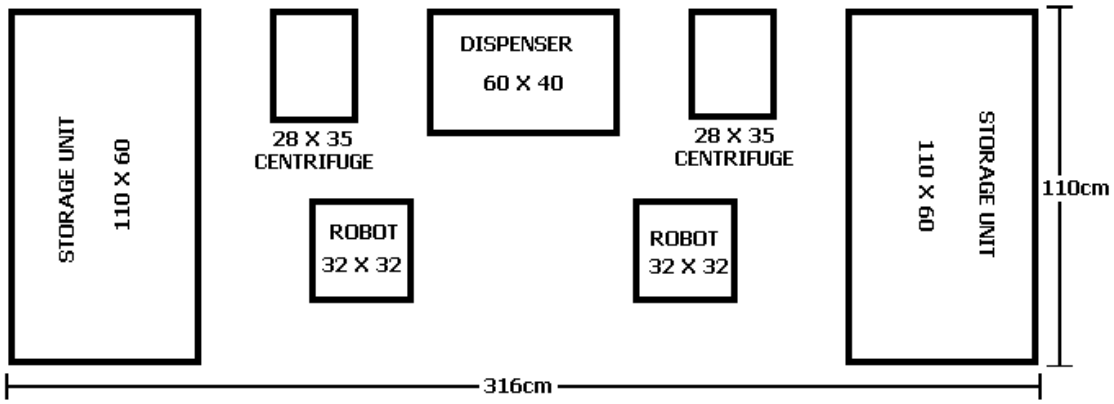

Figure 8.11 Configuration 1 layout

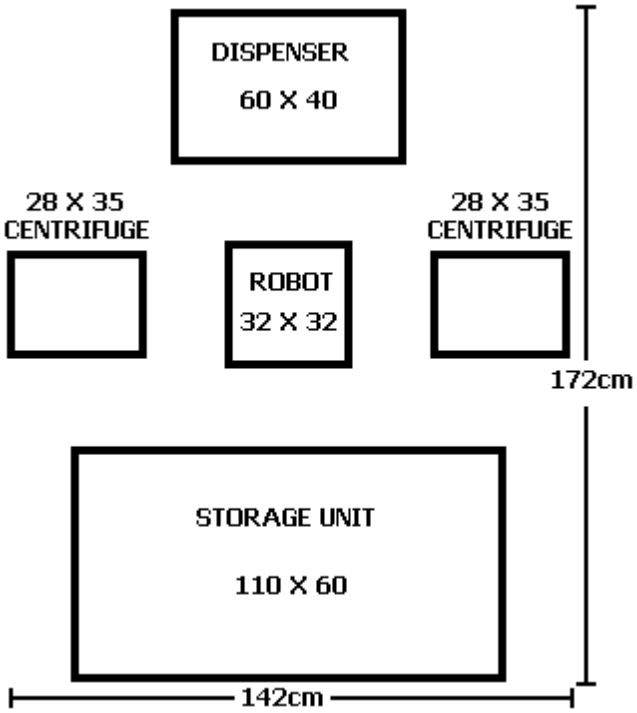

Figure 8.12 Configuration 2 layout

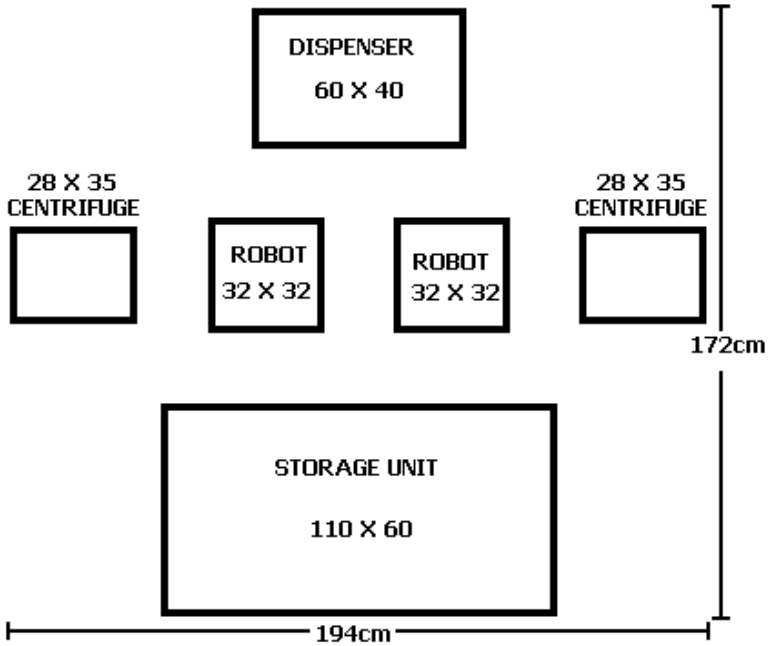

Figure 8.13 Configuration 3 layout

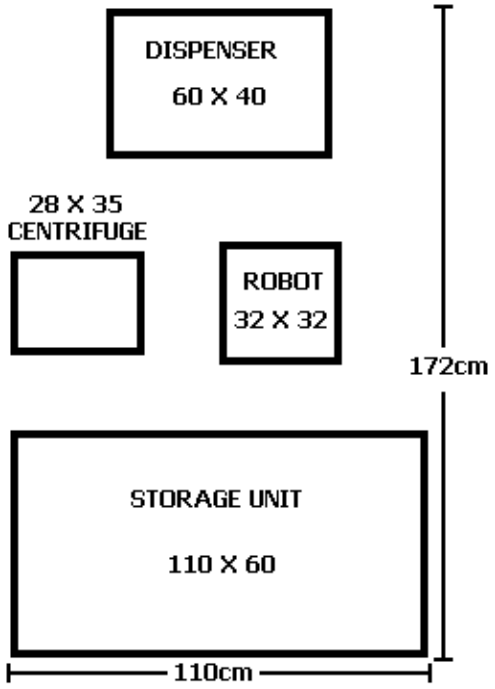

Figure 8.14 Configuration 4 layout

Let us now compute the variable cost per unit. Since the initial sample in this case is taken from cultured-cells its cost is neglected in this example: the cells are samples taken from patients that are grown to get the desired quantities. This is not always the case though.

Biological raw samples may sometimes be expensive. The price for an RNeasy mini kit is $227.00 and it contains 50 units. The variable cost for each unit is therefore $4.54.

The final factor missing is the price set per unit produced. For the purpose of calculating numerical values to compare the performance of the configurations a P = $6.00 is assumed. Recall that the value assigned to P does not affect the configuration selection process.

By (8.1) - (8.5) production line cost, variable cost, revenue and the time needed to return the investment can be calculated for the sake of comparing the configurations. Table 8.4 shows the cost results for each of the configurations.

Table 8.4 Results 4 configurations

Configuration	C_{PL} ($)	C_V ($)	R ($)	T (days)
1	126,761.20	2,928.30	441.79	135
2	68,866.00	2,374.42	765.58	90
3	123,700.80	2,928.30	941.79	132
4	64,630.00	2,424.36	779.64	83

Obviously $C_{PL4} < C_{PL2} < C_{PL3} < C_{PL1}$. If the only objective were to minimize cost then the best answer would be Configuration 4.

Configurations 2 or 4 with only one robot return the investment twice as fast as Configurations 1 or 3 that use two robotic arms. This is as the robots are by far the most expensive pieces of equipment used, constituting about 80% of the cost of the production line. Further analysis of the simulation results reveals that the utilizations of the robots in a two-robot configuration are much lower than that of the utilizations in the one-robot configurations. For a severe cost constraint scenario Configuration 4 appears to be optimal.

Table 8.5 Comparison results

Configuration	SR
1	0.0051
2	0.0076
3	0.0052
4	0.0082

The results obtained whenever using (8.6) to compare all the configurations appear in Table 8.5.

Configuration 4 maximizes the *SR* selection measure.

In cases of severe floor space constraint, Configuration 3 might be the best solution since it has the largest throughput combined with the smaller footprint.

If hypothetically money and footprint are not an issue, and the constraint is that the solution has to have two robotic arms (possibly dictated by other parts of the entire automation line) then the best solution may be neither Configuration 1 nor 3. Instead the best solution in such a case might be to build two Configuration 4 setups in parallel. The footprint of a double-configuration-4 is only one tenth larger; the cost of the production line increases by only 1%, however the throughput increases by 40%.

8.4. Example 2: Five-station-Process High-Throughput Implementation

The above configurations are plausible candidates for batch production systems. To explore a High-Throughput (HT) strategy solution to the same configurations some design adjustments have to be made. Most HT systems require customized tooling and other equipment. A common format of the arriving samples is that of a 96-well plate containing 96 samples (Figure 8.15); the centrifuge needs to be capable of running with these plates, and the dispenser should use a different tip capable of dispensing 96 wells at a time. This tooling and equipment changes necessitate a significant increase in cost (Figure 8.16).

Figure 8.15 96-well plate (QIAGEN, 1997)

There are four HT configurations that can be analyzed using the previous Arena® simulation setups. The main change with respect to the low-throughput simulations is that the centrifuge is ready to run with two entities (arranged symmetrically 180° apart), and each entity passing through the system represents a 96-well plate.

Figure 8.16 96-plate rotor for high-throughput centrifuge (QIAGEN, 1997)

Table 8.6 shows the throughput for an 8-hour shift for the various converted-to-HT configurations, and Table 8.7 shows typical prices of HT equipment.

Table 8.6 HT Arena® Simulation results

Configuration	Number of wells in 8-hours	Number of samples	Time per well
1-HT	966	92,736	30 sec
2-HT	764	73,344	38 sec
3-HT	966	92,736	38 sec
4-HT	762	73,152	54 sec

Table 8.7 High-throughput equipment cost

Equipment	Price ($)
Robot + Gripper	57,600
Centrifuge + Rotor	13,106
Dispenser	10,500
Storage Unit	1,200

Comparing Tables 8.1 and 8.6, reveal that entities throughput for HT implementation increases by almost 50%. This is due to the difference in the total queuing time. For low-throughput solutions, entities need to wait inside the centrifuge until all six slots are filled. In the HT case on the other hand since the rotor can only accommodate two entities, the total queuing time becomes significantly shorter.

The price for a QIAquick96 kit is $632.00 and it contains 4 plates. The variable cost for each plate is $158.00 or $1.65 per sample. The variable cost for an 8-hour shift (per day) for each

of the configurations is calculated using (8.2). The variable cost per day for Configuration 1-HT is C_{V1} = \$157,368.00, for Configuration 2 is C_{V2} = \$120,712.00, for Configuration 3 is C_{V3} = \$157,368.00, and for Configuration 4 is C_{V4} = \$120,396.00.

Modified cost of the production line and revenue for each of the HT configurations is shown in Table 8.8.

Table 8.8 High-throughput cost results

Configuration	C_{PL} (\$)	R (\$) (per day)
1-HT	158,356.40	399,048
2-HT	86,983.60	319,352
3-HT	158,227.20	399,048
4-HT	86,627.60	318,516

In this example we observe that the revenue does not significantly change from one configuration to another; whereas the cost of a two robot implementation is double that of a single robot implementation. In a high-throughput solution it appears best to always try to minimize the number of pieces of HT equipment.

So far in this chapter we discussed a specific application. In this example the robotic arm is the most expensive piece of equipment and it represents a high percentage of the total cost of the production line. This could not always be the case. There are other types of robots used in biotechnology, pharmaceutical and food applications. In particular there are also pick and place devices available which could perform just the loading and unloading of the centrifuge. The manipulation of the labware has in such a case to be done by other means like conveyors. Each choice has its advantages and disadvantages.

In biology and chemistry not all samples behave in the same way. It has to be taken into consideration that whenever dealing with live samples there is only an expected tendency. That does not mean that every single cell reacts exactly as the rest, not even mentioning that different samples are expected to behave differently. The samples might need different timing, and environmental conditions among others. If for other type of samples different centrifuge times are needed, the results might change drastically.

Issues related to the yield of the different protocols have not yet been explored. Not all protocols produce the same yield. It is very important to figure out the best way to add yield considerations to the optimality criterion. That would represent a more robust decision method.

8.5. Example 3: Complete RNA Purification Protocol

As illustrated before, one protocol with given set of sequenced operations can be automated using different configurations. Taking into account that the previous subset example that had only five different steps had four different possible equipment configurations, one can only imagine the endless number of possibilities that arise with a 17 steps protocol. For this reason, only the two extreme cases are shown as an example of the effectiveness of the method to decide between different configurations.

First, since the scope of this dissertation does not include the operations of labeling the samples, configurations that involve sharing of operation equipment are excluded from the study. To be able to share equipment, samples need to be labeled and tracked in order to make sure that all samples follow the correct step sequence. If the steps are not followed in the correct order, many reactions may not take place. As labeling steps are not included, and to ensure the correctness of the protocol, no operation equipment is shared and all steps are performed sequentially. Only the robotic manipulator may be shared.

The different configurations available for the simulation of the complete RNA purification protocol are drawn from the different methods of transporting the samples between stations. These configurations can for example include among others: pick-and-place devices for every station, pick-and-place devices combined with conveyors for the operations that allow them (Chapter 6), combination of sharing of robotic manipulators and conveyors, or one robotic manipulator to serve and share every station.

8.5.1. Candidate Configurations for RNA Purification

8.5.1.1. Configuration 1: Robotic manipulator mounted on a track rail

This first configuration was chosen because many available solutions, for example the High-Throughput Scripps System describe in Chapters 3 and 4, use this manipulation method. The Scripps System as we recall involves a single robotic manipulator that reaches every piece of

equipment and manipulates and transports all labware The Scripps High-Throughput System has 11 pieces of equipment situated around a central robotic manipulator. Some of which are stacked. We opted in this example not to discuss 3D solutions. For planar implementation a track rail on which the robotic arm is mounted is included in this simulation. This allows all 17 stations to fit around the robotic manipulator, increasing the manipulator's reach.

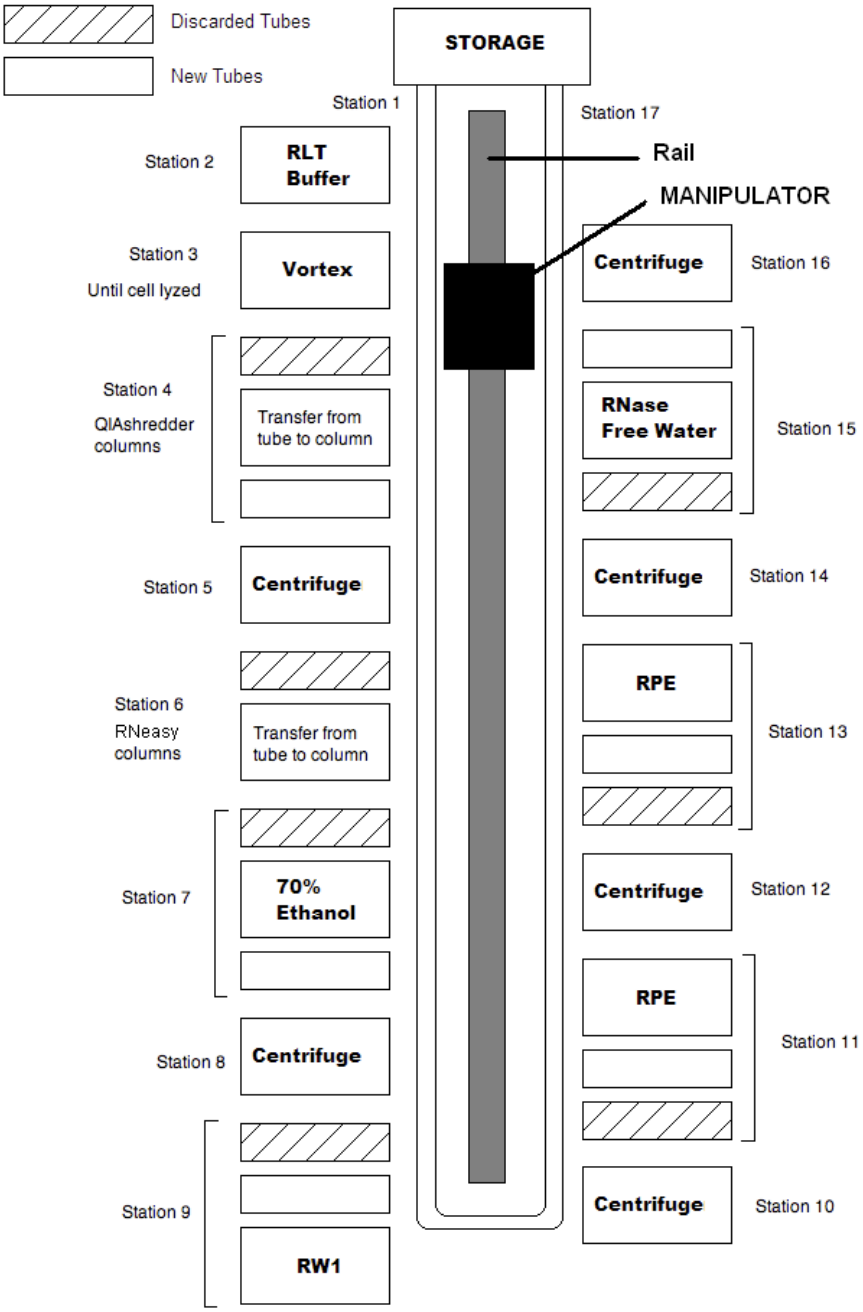

Figure 8.17 RNA Purification protocol Configuration 1 layout

Figure 8.17 shows the possible layout for Configuration 1. In Configuration 1 all 17 stations are placed around the track rail and the robotic arm in an oval configuration to ensure that the robotic manipulator can serve every station. The robotic arm performs all labware manipulation among stations in the system. The storage unit is shared between Station 1 and Station 17. In this case no labeling, tracking or protocol conflicts exist since the storage unit has two separate compartments, one where the cells are stored and from where these are taken out for processing, and the other one where the final RNA is stored at the end of the process.

The equipment used for Configuration 1 is: the equipment use for all operations plus the robotic manipulator and the track rail where it is mounted. For the operation this implementation method uses one storage unit, six dispensers, one vortex, two automatic pipettes and column stands, and six centrifuges.

8.5.1.2. Configuration 2: Combination of pick-and-place devices and conveyors

In the other possible extreme, we attempt to use the simplest possible manipulation method for each of the instruments in the line. This comprises of conveyors for any of the operations that allow use of conveyors, and simple pick-and-place devices for the instruments that absolutely demand it. Recalling Chapter 6, operations and equipment such as dispensers and automatic pipettes can be fed by means of conveyors and feeder mechanisms, whereas centrifuge and vortex definitely need pick-and-place devices. This configuration uses, other than the instruments used for the operations: one storage unit, six dispensers, one vortex, two automatic pipettes and column stands, and six centrifuges (the same as in Configuration 1), a conveyor belt that goes around all 17 stations, and 7 pick-and-place devices that manipulates the labware from the feeders mechanisms in the conveyors to the vortex and centrifuges. Each pick-and-place device serves one station and one instrument. Figure 8.18 shows the possible layout for Configuration 2.

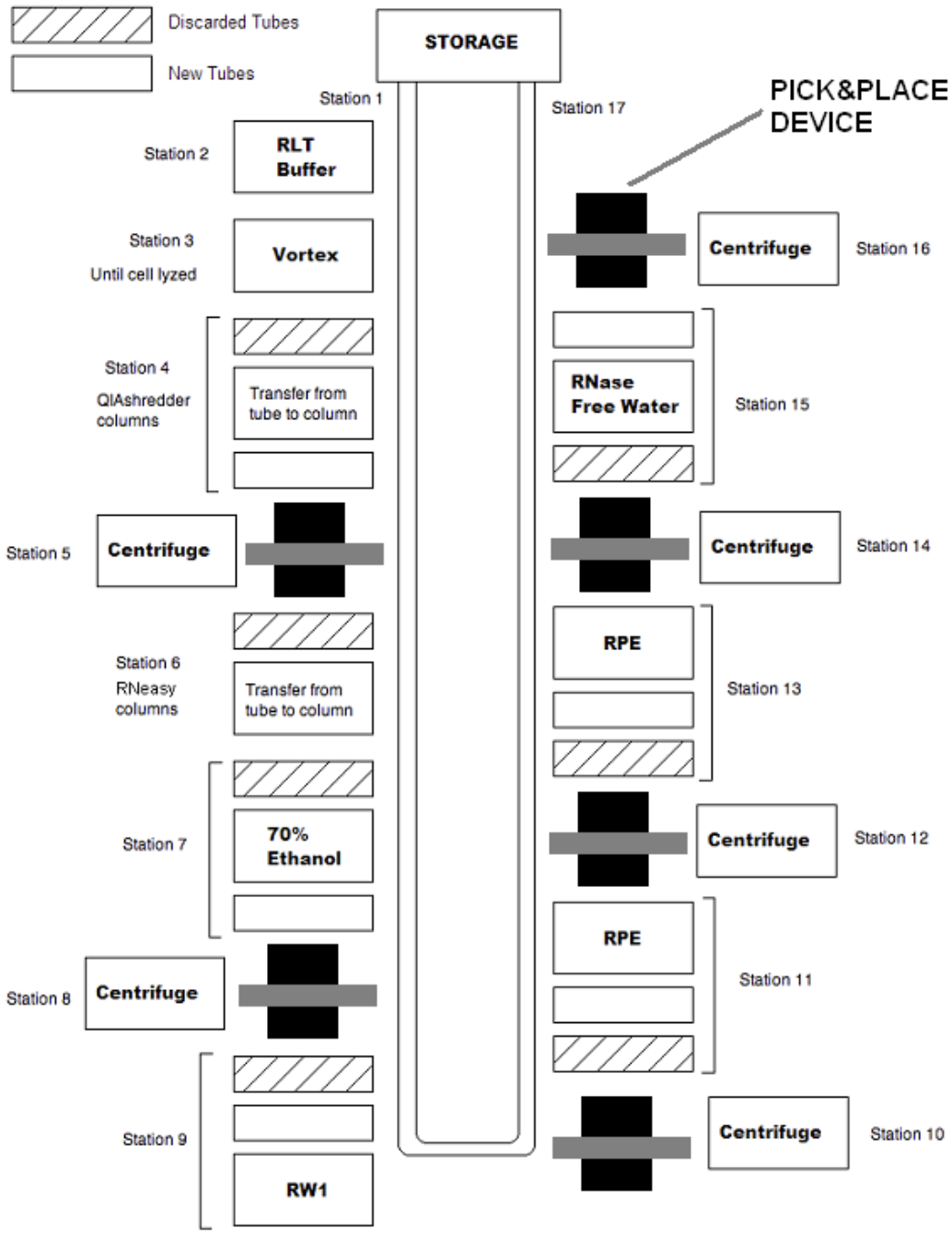

Figure 8.18 RNA Purification protocol Configuration 2 layout

8.5.2. Simulation with Arena® and Selection Ratio Configuration Selection

First, both configurations are simulated using Arena® software and the throughput of the system is recorded. This is the throughput that will be used for further calculations. Table 8.9 shows the results of the simulations for the total throughput for configuration 1 and 2 in an 8

hours shift, and the time that each sample takes. In both cases the samples entered the system every 30 seconds.

Table 8.9 Simulation throughput results

Configuration	Throughput (number of samples)	Time per sample
1	124	233 sec
2	234	124 sec

The following step is to calculate what each configuration costs. The equipment used for this validation is the same as the equipment used in the 5 steps example from the previous sections plus some additional equipment not considered in the previous example. Table 8.10 features the complete list of typical equipment prices needed for RNA purification that were used to calculate the cost of the configurations.

Table 8.10 Price of equipment used for RNA Purification protocol

Equipment	Price ($)
Robot	54,600
Track Rail	10,000
Pick-and-Place Handling System	7,800
Conveyor (price for 21in)	740
Centrifuge + Rotor	4,238
Vortex	1,200
Dispenser	2,500
Automatic Pipette and Column Stand	1,500
Storage Unit	1,200

In order to calculate the cost of each of the production lines, the total area for each of the configurations and the foot print value FP used for those calculations need to be obtained. The area is calculated in a straightforward manner from the diagrams of the layout of each configuration. Thereafter the area is multiplied by a 1.5 factor as explained in the example before. Table 8.11 shows the values for the dimensions results.

Table 8.11 Layout results for RNA Purification

Configuration	Peripheral Margin (cm)	Area (sq-ft)	FP value (sq-ft)
1	1488	109	163.73
2	1808	206	309.42

By observation of Tables 8.9 and 8.11 there are few conclusions that can be drawn before starting to use the automation configuration selection method. If the goal is to obtain the maximum throughput possible, Configuration 2 will be the choice because no time is wasted on waiting time for the manipulator. If on the other hand the goal is to minimize footprint, Configuration 1 is superior because by including the pick-and-place devices in Configuration 2 the footprint almost doubles. If there is a given space available and it turns out that Configuration 2 cannot be physically fit into that space then disregarding any other constraints, Configuration 1 becomes the only choice.

Assuming that there is not such space limitation constraint, the next step for continuing with the automation configuration selection method is to use the previous given information and (8.1) – (8.5), the cost of the production line, the revenue, and the number of days to return investment can be calculated for each of the configurations. Figure 8.12 shows these results.

Table 8.12 Cost results

Configuration	C_{PL} ($)	R per day ($)	T (days)
1	117,277.20	562.96	209
2	126,511.40	1,062.36	120

It is clear now that even though Configuration 2 has almost twice as much footprint and costs more, since it has a higher throughput the return on investment takes less time.

Finally the Selection Ratio criterion is calculated which confirms the selection based on return on investment time. These results appear in Table 8.13.

Table 8.13 Comparison results

Configuration	SR
1	0.0010
2	0.0018

These results are intuitive as the examples have been chosen to represent two extreme solution approaches to RNA Purification Protocol. As mentioned earlier, there are more possible solutions. If the configurations available for comparison are more closely related, the results might not be so straightforward without calculating the respective Selection Ratios. If, for example, there

is a space constraint that does not allow that each station has its own pick-and-place device, then these devices have to be shared to reduce footprint while compromising throughput.

8.6. Summary

This chapter examined the feasibility of a computer-aided automation design for biotechnology applications. It proposed a selection process that with the help of Arena® software the most appropriate automation solution for a given protocol could be evaluated and selected. The process used a selection criterion which included the production line cost, its work space and its throughput to select the best solution among the different configuration evaluated.

The design procedure follows three steps:

1. Choosing of plausible configurations for a manual protocol to be automated by means of educated deductions and personal experience of the designer. This includes the equipment to be used, the transportation means of the samples from station to station, and the actual space available for the given protocol. This has to be clear for each of the configurations that need to be evaluated.
2. Sketching of the layout of the different configurations, creating the Arena® model and simulation of the model for each configuration. This model has to take into account the resources used for each of the stations, the transportation means between stations, and the timing considerations of the protocol.
3. Calculation of the cost of the production line, the income and the revenue in order to calculate the Selection Ratio to be able to compare the different configuration and select the most appropriate one.

Application of the Selection Criterion may sometimes be done in an approximated manner. The hardware cost of each candidate configuration may be done by considering only the major most expensive pieces of equipment. The throughput, rather than being computer-aided computed, may be approximated using the relationships developed in Chapter 7. We recall that the approximated throughput calculation ignores the effects of finite transportation times.

9. GROUP TECHNOLOGY APPLIED TO BIOTECHNOLOGY

9.1. Introduction

Group Technology (GT) is based on the automation design notion that although parts may be different, there are also important similarities. These similarities can be associated with the attributes of the part or with the manufacturing process needed to produce the part (Groover, 1996) (Jaganathan, 2007) (Zhang, 2010).

In GT similar parts are identified and grouped together in order to take advantage of their similarities for design and production. Consequently many production cells in an automation line are designed to produce a family of parts rather than a single unique part, There is often a need for special production equipment in order to produce that family part (Zhang, 2010).

There is an important concept in GT known as Composite Part. Composite Part is a hypothetical part that includes all design and manufacturing attributes of the entire parts family. Automation systems that are capable of producing such composite parts are known as Flexible Automation Systems. The relation between part design features and manufacturing operations produces the Composite Part features. Conceptually the flexible automation line must include all machines required to make the Composite Parts and should allow the fabrication of parts with various sizes (Zhang, 2010).

There are three basic steps in group technology planning. First, a code must be created to allow for the needed part classification. Then, the parts must be classified into the different families and be represented each by a unique code that represents only one part type. Last, the layout of the various equipment and possible manufacturing cells is designed. It is worth mentioning that the starting point of GT implementation is a set of fully automated dedicated solution for each of the parts that need to be produced.

9.1.1. Code Representation and Part Classification

A GT code is an alphanumeric sequence of characters that should have the following three characteristics: (a), the code should be flexible enough to handle current as well as future parts, keeping in mind that a restrictive scope of the parts is included. (b), in order for a GT code to be useful it should discriminate the parts with different values for key attributes. (c), the code should uniquely identify each part and should fully describe the part from a design and a manufacturing point of view. An important issue to understand is that too much detail results in redundant information that is more difficult to handle. However too little detail may result in a useless code that is not capable of distinguishing parts apart.

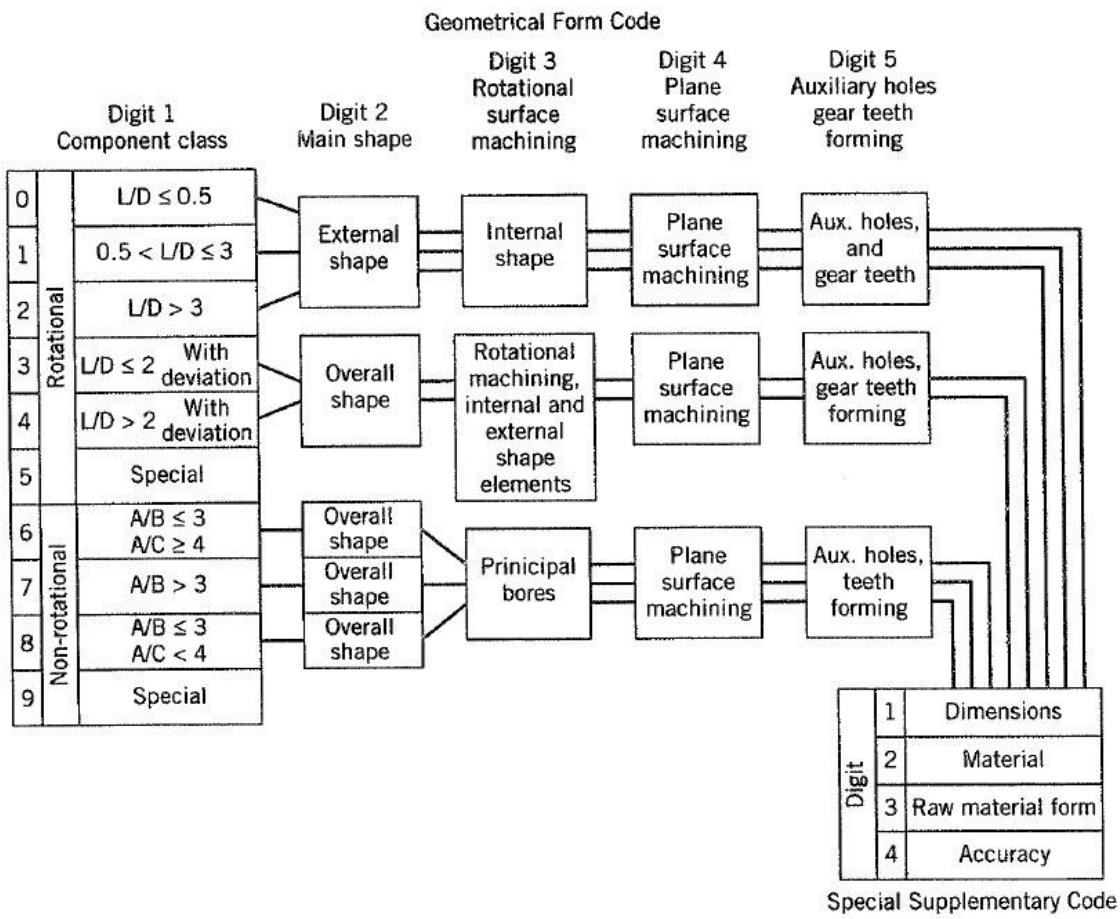

Figure 9.1 Example of OPITZ code (Askin, 1993)

An example of a well known code for implementing group technology philosophy in manufacturing is the OPITZ code (Girdhar, 2001). For better understanding of how a GT code looks like and the kind of information it holds, the OPITZ code is presented in Figure 9.1. For

example, this code has three sections: the first section, corresponding to the first five digits, codes the geometric information of the part. The second section has four digits and is the supplementary code. It contains information such as dimension material, original shape of the material, and accuracy. The third section, not necessarily utilized by all users, is a secondary four digit code defined by the user. This section may be used to describe production operations and sequencing information (Kanji, 2014) (Girdhar, 2001).

Once a proper code for the specific problem at hand has been defined, each part is assigned a unique code for identification purposes, and classified with in the corresponding group family. There are three common techniques to classify parts: description based on part design attributes, description based on part manufacturing attributes, and description based on both. For example, in mechanical assembly common part design attributes taken into account for part classification are: major dimensions, basic external shape, basic internal shape, and material type. Part manufacturing attributes that are commonly taken into account for part classification are: major process, operation sequence, machine tools, cutting tools, and tolerances. The classification and coding of the parts helps to develop the flexible automation flow analysis (Zhang, 2010).

9.1.2. Layout and Flow Analysis

The third step in GT planning is the layout design. This is carried out by doing a production flow analysis. Since at this point the purpose of this study is only to validate a plausible implementation of GT in biotechnology, and a reduced pool of protocols is used, the production flow analysis is not within the scope of this discussion. A brief explanation of how the flow analysis is done and why it is not applicable at this point is given.

One of the basic methods to perform a production flow analysis is to create a binary matrix with dimensions *i-by-j* where *i* is the number of machines and *j* is the number of parts. Such matrix interprets the binary entry 1 as a “machine used in the processing of the part”. The rows and columns (machines vs. parts) are often ordered using Rank-Order Clustering, and each cluster is then represented in the layout of the manufacturing cell (Jaganathan, 2007).

In this chapter we shall discuss only flexible automation that combines two protocols. Each biotechnology protocol has to be carried out sequentially and cannot be modified. Hence, the re-ordering of the matrix rows is not possible if one would attempt to use clustering techniques to perform the floor analysis. When including more protocols and sample types this might be a possibility depending on the relation between machines and products. And the different clusters could represent different manufacturing cells for each different sample/protocol family.

9.2. Introduction of Group Technology Implementation in Biotechnology

To demonstrate the feasibility of GT for biotechnology flexible automation the collection of protocols selected to illustrate it consists of processes for the preparation of RNA and DNA from different types of raw input samples, such as, different animal tissues, animal cells and plant cells. These protocols have been explained in detailed in Chapter 5 highlighting the similarities and differences. The main similarities and differences among these protocols are:

- Disruption of initial samples using lysis buffer is used for animal and plant cells for both protocols: RNA and DNA purification. The actual lysis buffer used for each of the samples type is different, but the procedure has the same two steps: first, addition of lysis buffer, then, vortex until the cells are lysed.
- Homogenization columns are used as homogenization methods for both purification of RNA as well as DNA from animal cells.
- In the case of plant cells, the use of homogenization for purification of RNA and DNA samples is carried out by means of homogenization columns. Although homogenization columns are used in both cases, the sequence of steps is not the same. In DNA purification prior to the transferring of the sample to the homogenization column a centrifuge step is added for better binding of DNA molecules to the membrane.
- For animal tissues, a mechanical disruption method is employed to disrupt and homogenize the sample for both, purification of RNA and purification of DNA.

- The actual isolation of RNA procedure becomes the same for the different types of initial samples once the samples have been disrupted and homogenized. This last remark is also true for DNA isolation.

Table 9.1 shows the steps of these six protocols already explained in Chapter 5. For better visualization of the differences and similarities among the protocols the protocols are written side by side.

The goal of this study is to demonstrate the feasibility of the use of GT for simplifying the flexible automation of biotechnology processes. To do so, we will follow the three known steps in group technology planning.

First, a code is defined. The code proposed and explained in this chapter, is not intended to be a complete.

Second, parts are classified. To demonstrate that biotechnology samples and protocols can actually be classified into families, we illustrate that this collection of protocols can either be classified into groups based on part manufacturing attributes and also based on part design attributes. We also demonstrate that each sample of the family can be uniquely coded using the proposed code and each sample can be set apart from another.

Third, a proposed equipment layout for the system based on part design attributes is presented. The proposed layout is drawn from the Composite Part of the sample/protocol family.

After the complete group technology planning is implemented for the system based on part design attributes, we validate the flexible solution by simulating the system with help of the Arena® simulator and it is compared to automated dedicated systems for single sample/protocol.

Table 9.1 Case study protocols – Preparation of RNA and DNA from animal tissue, animal cells, and animal plant cells

<table>
<tr><th colspan="3">RNA</th><th colspan="3">DNA</th></tr>
<tr><th>Animal Cells</th><th>Plant Cells</th><th>Animal Tissue</th><th>Animal Cells</th><th>Plant Cells</th><th>Animal Tissue</th></tr>
<tr><td colspan="2">Addition of lysis buffer to cells in 2ml tubes
Note: Different lysis buffers are used for different cells</td><td>Addition of stainless steel bead to tissue sample in 2ml tube</td><td colspan="2">Addition of lysis buffer to cells in 2ml tubes</td><td>Addition of stainless steel bead to tissue sample in 2ml tube</td></tr>
<tr><td colspan="2">Vortex until cells are lysed</td><td>Addition of lysis buffer to tissue sample in the 2ml tube</td><td colspan="2">Vortex until cells are lysed</td><td>Addition of lysis buffer to tissue sample in the 2ml tube</td></tr>
<tr><td rowspan="4">Transfer lysate into homogenization column</td><td rowspan="2">Transfer lysate into homogenization column</td><td>Place the tissue sample in the device (shaker) adaptor</td><td rowspan="4">Transfer lysate into homogenization column</td><td rowspan="2">Centrifuge</td><td>Place the tissue sample in the device (shaker) adaptor</td></tr>
<tr><td>Shake</td><td>Shake</td></tr>
<tr><td rowspan="3">Centrifuge</td><td>Rotate adaptor</td><td rowspan="2">Transfer supernatant into homogenization column</td><td>Rotate adaptor</td></tr>
<tr><td>Shake</td><td>Shake</td></tr>
<tr><td rowspan="4">Centrifuge</td><td>Centrifuge lysate</td><td rowspan="4">Centrifuge</td><td>Centrifuge lysate</td></tr>
<tr><td rowspan="3">Transfer supernatant to a new 2ml collection tube</td><td>Transfer supernatant to an homogenization column</td><td rowspan="3">Centrifuge</td><td>Transfer supernatant to an homogenization column</td></tr>
<tr><td>Centrifuge sample</td><td>Centrifuge sample</td></tr>
<tr><td colspan="3">Addition of ethanol to the homogenized lysate</td><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Transfer the sample to a spin column placed in a 2ml tube</td><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td><td colspan="3">Incubation</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td><td colspan="3">Centrifuge</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td><td colspan="3">Incubation</td></tr>
<tr><td colspan="3">Addition of elution buffer to spin column</td><td colspan="3">Centrifuge</td></tr>
<tr><td colspan="3">Centrifuge</td><td colspan="3"></td></tr>
</table>

9.2.1. Four Digit Code Representation for RNA and DNA Isolation

A four digit code that follows the characteristics of a GT code is defined to describe the collection of these six protocols. This code is use to validate GT implementation in biotechnology (Cardona, 2014b).

The physical attributes as well as the process attributes of the samples are recognized by understanding the different protocols and by taking educated guesses based on prior experiences. The different parts of the code can be recognized by inspecting similarities and identifying the common processes. In this example, the common attributes are the initial samples: animal cells, animal plants, and various animal tissues; the processes: RNA purification and DNA purification; and the possible product outcomes: large volume or large concentration.

The first two attributes, the type of initial sample and the process needed, are obviously a must. The third attribute, possible product outcomes, refers to a choice that has to be made between having a large concentration of the final product or a large volume of the final product. For such collection of protocols, there is a tradeoff between volume and concentration. As the final steps of the protocols are different if aiming for large volume than if aiming for large concentration, the automation solution will also be slightly different. This attribute reflects directly on the automation solution, and therefore has to be included in the code.

As a result, the four digit code is as follows:

- The first digit indicates the type of input sample which can be one of three: animal cells, animal tissues, or plant cells.
- A second digit is used only for the description of structural characteristics. It is only used to distinguish between the different possible animal tissues. These animal tissues can be of three types: fatty tissues, fibrous tissues, or other tissues.
- The third digit describes the process that is carried out. It can either be, RNA, DNA or proteins.
- The fourth digit indicates if the product should have a large concentration or a large volume.

Table 9.2 presents the code proposed for the implementation of group technology for biotechnology protocols.

Table 9.2 Code use for Group Technology implementation

First Digit:		Second Digit:		Third Digit:		Fourth Digit:	
(Input sample)		(Structural characteristic)		(Process)		(Product Outcome)	
Animal cells	1		0	RNA	1	Volume	1
Animal Tissue	2	Fatty Tissue	1	DNA	2	Concentration	2
		Fibrous Tissue	2	Proteins	3		
		Other Tissue	3				
Plant cells	3		0				

9.2.2. Part Classification

The classification of parts for biotechnology protocols can be based on part design attributes and part manufacturing attributes. Both possibilities are explored in this section.

9.2.2.1. System Based on Part Manufacturing Attributes

Table 9.3 Purification of RNA from different sources

RNA	
Animal Cells, Bacteria, Yeast	**Plant Cells**
Addition of lysis buffer to cells in 2ml tubes Note: Different lysis buffers are used for different cells	
Vortex until cells are lysed	
Transfer lysate into homogenization column	Transfer lysate into homogenization column
Centrifuge	Centrifuge
	Transfer supernatant to a new 2ml collection tube
Addition of ethanol to the homogenized lysate	
Transfer the sample to a spin column placed in a 2ml tube	
Centrifuge and discard flow-through	
Addition of wash buffer to spin column	
Centrifuge and discard flow-through	
Addition of wash buffer to spin column	
Centrifuge and discard flow-through	
Addition of wash buffer to spin column	
Centrifuge and discard flow-through	
Addition of elution buffer to spin column	
Centrifuge	

Preparation of RNA from various candidate samples can help demonstrate how GT can be implemented to design manufacturing lines for biotechnology protocols based on part manufacturing attributes. The procedures used for purification of RNA from animal cells are the same as for RNA extracted from bacteria or yeast. It also has quiet a few number of operations in common with Purification of RNA from plant cells. These protocols are found in Table 9.3. Whenever implementing part classification based on part manufacturing attributes the parts are classified into families according to the characteristics of the manufacturing procedures. In this case, as these processes have common characteristics such as same equipment, same operations, and same sequences, they can be classified as one family.

Purification of RNA from these two initial samples, animal cells and plat cells, could be carried out using almost the same equipment and the same production line without needing to change the equipment configuration. What differentiates one protocol from another are the reagents used and the timing parameters. The lysis buffers for each of the samples is different and the first centrifugation for plant cells takes longer than the one for plant cells. Evidently the use of GT is very beneficial in this case as although we have different initial samples, and get different products (that is RNA from different samples), it is possible to group all of these samples into the same family as these have the same manufacturing attributes and share the same protocol.

Each sample can be uniquely coded using the proposed four digit code.

- 1011 – RNA from animal cells at high volumes
- 1012 – RNA from animal cells at high concentration
- 3011 – RNA from plant cells at high volumes
- 3012 – RNA from plant cells at high concentration

9.2.2.2. System Based on Part Design Attributes

GT based on part design attributes approach could also be applied to biotechnology. Table 9.4 shows two distinct protocols: the left side is preparation of RNA form animal cells, and the right side is the preparation of DNA from the same animal cells. These two protocols could be classified into the same part family because of the part design attributes. As mentioned before, in

mechanical assembly common part design attributes taken into consideration for part classification are major dimensions, basic external shape, basic internal shape, and material type. In the case of biotechnology these physical characteristics could be interpreted and extrapolated in various ways, as the part, which in this case would be the sample, changes its characteristics throughout the process. For example, a wooden part will be made of wood throughout the process, meaning the raw material is wood and the final product is a part made of wood. In biotechnology, and specifically in this case, the raw material is animal cells and the final product could be RNA or DNA molecules, thus changing its physical characteristics. One way to understand classification based on part design attributes is to look the characteristics of the initial sample. As the same animal cells are used for both protocols, all samples have the same internal and external shape and same initial sample, thus being able to classify them within the same part family.

Table 9.4 RNA and DNA preparation from tissue-cultured cells

RNA	**DNA**
Animal Cells	**Animal Cells**
Addition of lysis buffer to cells in 2ml	
Vortex until cells are lysed	
Transfer lysate into homogenization column	
Centrifuge	
Addition of ethanol to the homogenized lysate	Addition of wash buffer to spin column
Transfer the sample to a spin column placed in a 2ml tube	Centrifuge and discard flow-through
Centrifuge and discard flow-through	Addition of wash buffer to spin column
Addition of wash buffer to spin column	Centrifuge and discard flow-through
Centrifuge and discard flow-through	Addition of elution buffer to spin column
Addition of wash buffer to spin column	Incubation
Centrifuge and discard flow-through	Centrifuge
Addition of wash buffer to spin column	Addition of elution buffer to spin column
Centrifuge and discard flow-through	Incubation
Addition of elution buffer to spin column	Centrifuge
Centrifuge	

In these two protocols an AllPrep DNA/RNA mini column available from QIAGEN Inc. is used (QIAGEN, 2008a). The AllPrep column has chemical properties that make the genomic DNA bind to the residue of the column. This means that genomic DNA can be eluted directly from

the membrane of the AllPrep column, and the RNA can be obtained from the flow through of the AllPrep column.

Taking into consideration preparation of RNA and DNA from animal cells (Table 9.4), it can be observed that the protocols have some common operation, but preparation of DNA has fewer steps than the preparation of RNA. In this case part classification is done using a system based on part design attributes. Although the processes are not exactly the same, they use the same initial sample for RNA and DNA preparation from the same animal cells. In other words these parts could be classified as one family.

In this example, each part can be identified by a unique code using the four digit code representation described in Section 9.2.1 and the system can handle four different parts:

- 1011 – RNA from animal cells at high volumes
- 1012 – RNA from animal cells at high concentration
- 1021 – DNA for animal cells at high volumes
- 1022 – DNA from animal cells at high concentration

9.2.3. Equipment Layout

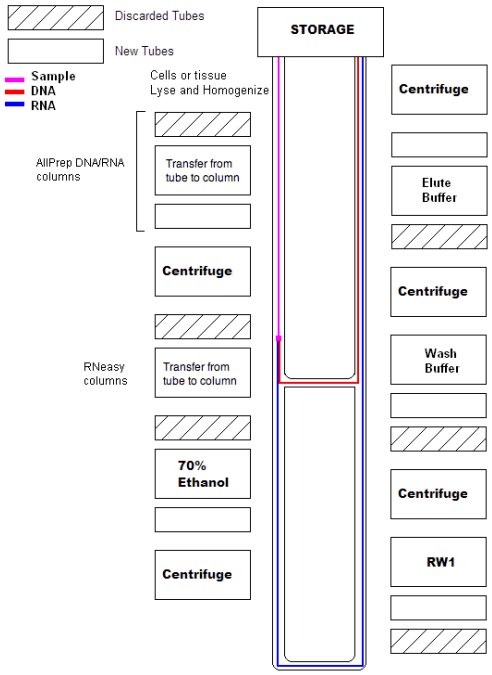

Figure 9.2 Conceptual design for the preparation of DNA and RNA from animal cells production line

Using the Composite Part concept for the RNA and DNA purification problems and by carefully inspecting the protocols, an educated guess for an appropriate layout of the equipment can be made. The modified conceptual design of the production line for the preparation of RNA and DNA from animal cells applying Group Technology is shown in Figure 9.2.

9.2.4. Validation

The example was evaluated using Arena® Simulation Software. The different Arena® modules were built employing the Automation Modules described in Chapter 6.

The impact of implementing GT in biotechnology was evaluated for two scenarios:

In the first scenario we compared the dedicated line that produces only RNA from animal cells (already evaluated in Chapter 8 as Configuration 2), to the flexible line proposed in Section 9.2.3 which is capable of producing both RNA and DNA from animal cells. For the sake of comparing “apples to apples”, in the first scenario although the GT configuration is a flexible solution, in the simulation the line was only producing RNA from animal cells.

The results obtained for the simulations in scenario 1 are presented in Table 9.5.

Table 9.5 Results comparing a dedicated production line against a flexible line producing only RNA from animal cells

	Dedicated Production Line	Flexible Production Line
Samples (code)	1011	1011
Throughput for an 8h shift	234	183
Process time per sample (s)	124	157
Peripheral Margin (cm)	1808	2531
Area (sq-ft)	206	288
FP value (sq-ft)	309	433
CPL ($)	126,152	187,063
R per day	1062	830
T	120	225
SR	0.0018	0.0009

The two production lines in scenario one are compared using the selection criteria developed in Chapter 8. As can be seen in Table 9.5, and is intuitively clear: from the selection ratio criteria SR, the dedicated production line is a better choice for producing only one type of sample at the time.

The dedicated line produces almost 30% more samples in a shift than the flexible automation solution. This is because as the flexible line was designed using the Composite Part concept, samples have to go through 14 more stations even though the samples are not processed by those extra stations. So the path the samples cover in the flexible line is longer than the one in the dedicated line taking more time.

The floor footprint in the flexible line is almost 50% larger than the footprint of the dedicated line as for the flexible line the equipment and stations for the preparation of DNA are included. This also means that the cost of the production line is increased by about 40%.

We can clearly deduce from this results that implementation of GT to include flexible automation to biotechnology protocols clearly have its tradeoff. More flexibility means, higher costs and a slight reduction in throughput. On the brighter side, a 40% increase in costs only had a 30% decrease in throughput. All of these considerations need to be taken into account by the designer when deciding if flexible automation is needed.

In the second scenario we compared the performance of having two dedicated production lines, one for RNA from animal cells at high volumes and the second for RNA from animal cells at high concentration, against a the flexible line producing these same two samples. The results for scenario 2 are presented in Table 9.6

Table 9.6 Results comparing two dedicated production line, one producing 1011 and the other one 1012, against a flexible line producing 1011 and 1012

	2 Dedicated Production Line	**Flexible Production Line**
Samples (code)	1011 and 1012	1011 and 1012
Throughput for an 8h shift	468	183
Process time per sample (s)	61	157
Peripheral Margin (cm)	3616	2531
Area (sq-ft)	412	288
FP value (sq-ft)	618	433
CPL ($)	252,304	187,063
R per day	1062	830
T	120	225
SR	0.0018	0.0009

In scenario two the selection ratio criteria SR also tells us what we expected, that two dedicated lines will have a better performance that one flexible line producing the two types of samples. However for this scenario the significant results go beyond that.

It is true that that the two dedicated lines can produce more than twice as much as the flexible automation line. However the question that the designer has to ask is under what cost this is possible. Obviously using two dedicated lines means having twice as much equipment, footprint, and costs. Three main issues have to be considered whenever choosing the right solution in this case. First, is the solution footprint within the available space of the facility? Is it the cost of duplicating every single piece of equipment within the budget available to design and development of the line? And last and more important, is the line meeting TP requirements. For example, if the flexible automation meets the throughput requirement there is no need to have two dedicated lines solution and even though the selection criterion says otherwise, the best solution is the flexible line.

9.3. Concluding Remarks

The benefits that come with the use of GT are many, to name a few: It promotes standardization of tooling and steps; it simplifies the production scheduling; manufacturing lead time is reduced; process planning is simplified; and this will all result in higher quality work. Group Technology also comes with disadvantages. The main two are: the production machines have to be rearranged, and identification of a part family could be complicated.

As already stated, this study only discusses the feasibility of using Group Technology for automation of Biotechnology protocols. The code presented only represents a subgroup of all the protocols available in Biotechnology and a subgroup of samples (QIAGEN 2008a). Once the feasibility has been demonstrated, now the challenge is to describe a code that actually encompasses a larger family of protocols and samples.

The production flow analysis also has to be explored in more detail to be able to fully implement group technology in Biotechnology automation and have truly flexible laboratories that can perform a wider variety of protocols.

10. CONTRIBUTIONS AND FUTURE DIRECTIONS

10.1. Contributions

The key contributions of this dissertation are in the following categories:

Integration of LUOs

By carefully studying the typical and most recurring methods of implementing the twelve LUOs used in biotechnology, we created a classification based on automation integration architectures. This was named Automation Modules, AM, which are the building blocks for any biotechnology automated solution. The AMs were validated via Arena® simulations and are configurable blocks suitable to implement a variety of operations.

The above set of AMs cannot be considered an exhaustive set that will meet the needs of every single protocol. We rather present a methodology of how to construct such building blocks for operations that may not fit the mold of the already existing AMs. New AMs may be added in the future to seven AMs that we have proposed. Likewise, complex multi-purpose integrated equipment was addressed with the same design approach.

Some of the LUOs have been expanded in scope to include several additional odd automation aspects. These operations include manual operations that may be needed in the automation line but are not performed so frequently such as restocking of labware dispensers, replenishment of reagents and compounds, and mid-process equipment interchange. It also includes automated operations that were overlooked before which are vital for an automation line and the control of the line such as labeling operations and monitoring.

Automation Lines Timing and Structural considerations

Two strategies are formulated to approach the problem of biological and chemical samples that may exceed their maximum allowable shelf times.

The first strategy involves a Queue Clearing strategy and it is based on determining critical stations that may otherwise violate maximum allowable shelf times. Periodic stopping and

resuming times for the system input are calculated to ensure that the queue size at each critical station is kept within the given constraints. If the critical station satisfies the requirements, all other stations will do as well, as those other queues grow at a slower rate than the ones in the critical station.

The second strategy involves the Parallelizing of critical stations. A critical station is identified and is enhanced by adding an additional parallel unit to help the station process more parts during the same station's service time. The parallelizing process is done iteratively, identifying at each step the current most critical station. The iterations continue until all queues are within specs, or until terminating conditions occur. The terminating conditions that need to be included to obtain a feasible result are maximum number of units per station, budget limit, and footprint limit. The parallelization strategy in some cases may not converge into a valid solution and clearing strategy parameters need to be augmented. The parallelizing strategy may be employed in a context that is different than shelf time limitation – simply as a method to increase the system throughput.

A MATLAB® simulator was developed to evaluate the queue clearing and parallelization strategies formulated in this dissertation. The simulator used none of the formulas developed in this dissertation. It rather mimicked the automation line basic parts transfer and processing operations, and basic counting of parts going to and emerging of each station. The simulator was validated using Arena® software, comparing the results obtained from the MATLAB® simulator to the ones obtained with the Arena® software for various models, establishing that the neglecting of transportation times in the case of multi-manipulator implementation are justified.

Criteria for Automation Line Sub-System Implementation Selection

A return on investment approach established an economic model created to assess the multi-objective system configuration selection problem involving throughput, floor space, and cost constraints. Application of the economic model on candidate configurations of a subsystem and simulation of such subsystems to obtain their throughput via Arena® software, and a Selection Criterion based on the ratio between the throughput and the configuration-dependent construction

and operation cost was defined. This Selection Criterion helps identify the best solution among the candidate configurations for a given system.

These candidate configurations results from the implementation of the Automation Modules concept to a biotechnology protocol. Then, the candidate configurations are ensured to be within the given requirements and conditions of the protocol by using either Queue Clearing or Parallelization strategies. A full description of a computer-aided automation design for biotechnology applications using Arena® Software and the developed MATLAB® simulator has been completed.

Introduction to flexible automation

By using an illustrative example of a family of RNA and DNA purification protocols, we demonstrated the feasibility of applying Group Technology (GT) for introducing flexible automation to biotechnology protocol. Even though the collection of protocols and samples used in this study was very limited, it illustrates how the various elements of GT can become applicable to biotechnology automation. It also illustrates how biotechnology automation could benefit from GT implementation

10.2. Future Directions

The study should be extended to include protein purification and recrystallization protocols. Only off-the-shelf equipment is considered in the study, and the protocols demonstrated in this dissertation present only a small sample of a very large pool of available protocols. The dissertation focused primarily on RNA and DNA purification protocols. Automated protocols for protein purification and recrystallization are a clear future direction with regards to development of automation design guidelines. Some biotechnology automation advances depend on better understanding of the manipulation of proteins.

Inclusion of labeling steps would allow tracking of parts, monitoring and equipment sharing for a more efficient integration. One of the assumptions made throughout the dissertation was the avoidance of labeling steps. In the past, labeling operations have been left out of the group of twelve basic LUOs. In this study, labeling operations were discussed very briefly in Chapter 6 with regard to the implications of leaving them out of these set of operations

and the importance of including it as part of one of the LUOs. Labeling of samples would bring many great advantages and improvements to the automation system from different aspects. It would allow the development of strategies for equipment sharing, and for branching control the samples flow through the system. It would also allow the development of a feedback and monitoring system to control the progress of the protocols reactions. So far the methodology described in this dissertation only supports the sharing of the labware manipulation equipment as done via the management software that integrates, communicates, and controls all the equipment involved in the system. Going one step further, and being able to perform different steps in the protocol with the equipment of one station, could potentially have two desired results: reduced cost and reduced footprint. This will not necessarily always as extra stations and steps will be needed to be added for the purpose of writing and reading samples labels.

Inclusion of monitoring operations is vital for quality control throughout the complete automation line. A future direction that goes together with the inclusion of labeling steps is the inclusion of monitoring steps. In this dissertation we assumed that the operation of each automation line station is done in open loop using no feedback sensors to monitor each station's yield. Whenever aiming at production lines design for better quality control and minimal waste the inclusion of monitoring operations is a must. The challenges with including monitoring in biotechnology protocols in every step are mainly bi-fold various mid-products reactions may be problematic to evaluate; and the measuring of the quality of the sample may involve inclusion of foreign substances. Such test samples cannot be allowed to return to the process. One of the design considerations therefore has to be about the frequency of measuring the samples. A second consideration involving labeling of samples is about branching the automation line stations to paths that depend on the testing results.

The algorithms developed to accommodate automation line timing constraints neglected the transportation time between stations. It was assumed that since the transportation time is typically much smaller than the service time of the stations, the transportation time could be neglected. This is not necessarily so whenever, for instance, a single manipulator serves multiple stations. The queue clearing and station parallelizing algorithms need

to be extended to include transportation times. Likewise the MATLAB® simulator needs to be extended. It will also be interesting to explore the idea of increasing the effective service time of some upstream stations from slower stations via the delaying or manipulating of the transportation time. In other words, slowing down of the robots or the conveyors is equivalent to the addition of more queue space to each station, a practice that is implemented in part in The Scripps Research Institute uHTS platform.

For convenience of predicting the progression of the samples and of the programming of the MATLAB® simulator, parallel stations were assumed to run in a synchronized manner. Doing this has two direct results in the simulation and the prediction of the system throughput and other features of the automation line. First, the effective arrival period of a given station may be increased as the station now has to await a multiple of *Np(p)* parts to arrive at the station in order to be processed. Second, queues downstream of the parallel station tend to increase compared to queues that develop past a single station. This may conflict with *MQt(p)* or *MQs(p)* of following stations. By introducing asynchronous parallelization both issues may be resolved as each station will run independently from each other maintaining their resulting arrival periods and queues. To be able to fully implement asynchronous parallelization, an option for automation line branching needs to be included.

Throughput enhancement and yield enhancement are important follow up directions from this dissertation. As mentioned earlier a Parallelization strategy often results in an increased system throughput. This dissertation stopped short of exploring the combination of station parallelizing and system queues clearing strategies, left as an open future problem. The inclusion of monitoring steps to enhance the yield of each station surely impacts stations service times and consequently queues size. Exploration of how yield enhancing and system timing are related is left as future problem.

An extended study on implementation of Group Technology for flexible biotechnology automation is a future works which will introduce multiple system capabilities. In this dissertation we demonstrated the feasibility of implementing Group Technology in biotechnology automation. The benefits that come with the use of GT are many, to

name a few: It promotes standardization of tooling and steps; it simplifies the production scheduling; manufacturing lead time is reduced could be reduced as the production line would be ready to process any incoming part; process planning is simplified; and this will all result in a more dynamic and effective production line. GT also comes with disadvantages. To mention two: the production machines have to be rearranged to accommodate flexible automation, and the initial identification of a part family to perform the GT implementation could be complicated. The benefits from it surpass the disadvantage and this is why it is worth dedicating a study to it. In this feasibility study we only touched the tip of the iceberg. Possible future direction on GT for biotechnology includes the development of a full code that actually encompasses a larger family of protocols and samples. The code (as in mechanical assembly GT) must describe not only the samples but also the manufacturing processes. The needed LUOs and AMs must be inferred from the code. Production flow analysis has to be explored in more detailed to be able to fully implement GT in biotechnology automation, which as mentioned earlier involves sample labeling and automation line branching.

3D solutions could be an interesting solution for the space constraints involve in biotechnology. Finally, in this dissertation we only considered 2D automation solutions and the use of simple manipulators (which included conveyors, robotic arms, and pick-an-place devices). The methodology could be validated for 3D solutions and include customized manipulators such as reconfigurable robots and vertical track-based systems.

APPENDICES

APPENDIX A

FULL DETAILS OF THE PROTOCOLS OF CHAPTER 5

Purification of Total RNA

The complete protocol for purification of total RNA is divided into three main processes: disruption of the input sample, homogenization of the sample, and isolation of RNA (Figure 1).

Even thought the protocol is called Purification of Total RNA, only molecules larger than 200 nucleotides are purified. As a result, the homogenized sample has an enrichment of mRNA.

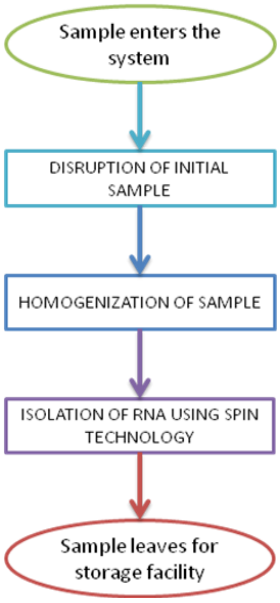

Figure 1. Process flow chart for Purification of RNA using spin technology

Purification of Total RNA from Animal Cells

Disruption Method

Disruption of initial cell samples consists of lysing the cell membranes by adding a Lysis Buffer to the sample originally contained in a 2ml tube and the vortexing or shaking of the tube until cells are lysed (Figure 2). At this point, the sample in the test goes from a clear color to a milky one. For a scientist performing this protocol manually, this is how a success of the lysing operation is determined.

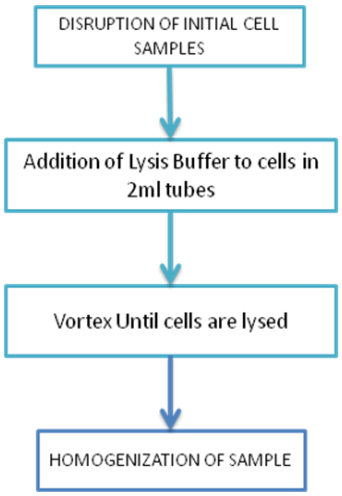

Figure 2. Disruption method for Animal Cells: Lysis Buffer and Vortex

Homogenization method

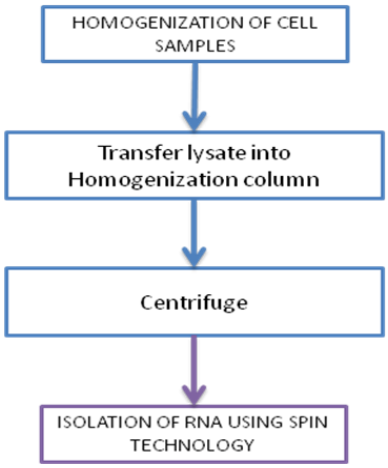

Figure 3. Homogenization method for Animal Cells: Homogenization column and spin technology

The homogenization method for animal cells consists of two steps: first, the lysate is transferred to a homogenization column by pipetting it out of the 2ml collection tube and into the homogenization column; then, the column attached to a 2ml collection tube is placed into a centrifuge and is spun for a certain period of time to force the entire contents through the column's membrane, which in this case shreds the bigger molecules and particles in the lysate (Figure 3). The result is a uniform sample obtained in a 2ml collection tube used for further isolation of RNA.

Isolation of RNA Using Spin Technology

The protocol employed for RNA isolation uses column chromatography combined with spin technology (Figure 4).

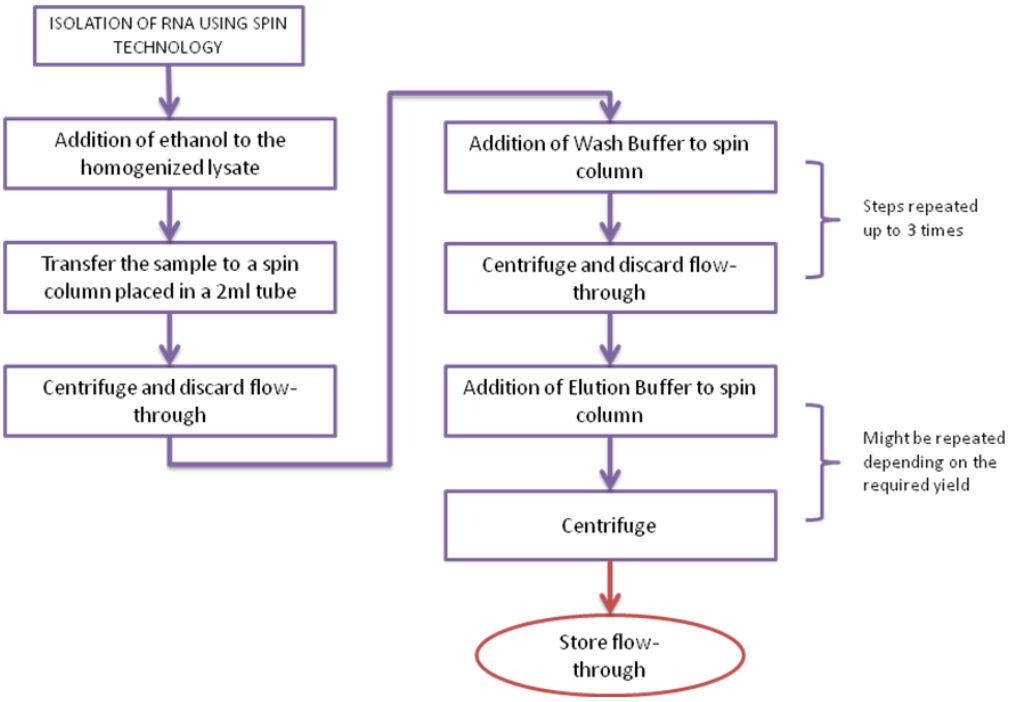

Figure 4. Isolation of RNA

First, ethanol is added to the 2ml collection tube containing the homogenized lysate sample. As mentioned above, this addition is done to create the conditions to promote binding of RNA to the membrane. The sample is mixed by pippeting in and out the liquid, making sure that the complete volume resides in the collection tube at the end of this procedure. Second, the sample is transferred to a spin column attached to a 2ml collection tube. Third, the sample is centrifuged for a given period of time, and the flow-through is discarded. Fourth, the wash buffer is added to the spin column. The wash buffer, due to its chemical properties, helps to wash off undesirable particles from the membrane by weakening the binding of those undesirable particles to the membrane. Fifth, the spin column is centrifuged and the flow-through discarded one more time. Steps 4 and 5 are repeated two more times with a different wash buffer to ensure that all undesirable particles and molecules are washed out of the membrane in order to have only the total RNA bound to the membrane before elution. Sixth, RNase-free water is added to the spin column as an elution buffer. Finally, the column is centrifuged again in order to obtain the Total RNA in RNase-free water in the collection tube. Depending on the desired yield, elution steps can

be repeated by using a second volume of RNase-free water or the previous flow-through. If a second volume of RNase-free water is used, the yield will be greater but at a lower concentration. If the flow-through is used for the second elution, the yield will be lower but at a higher concentration. The choice depends on what the sample requires and involves some cost considerations.

Purification of Total RNA from Animal Tissues

Purification of RNA from animal tissues assumes that the sample has been previously properly harvested, stabilized, and stored, and it is ready for RNA purification without any other pre-processing.

Disruption and Homogenization methods

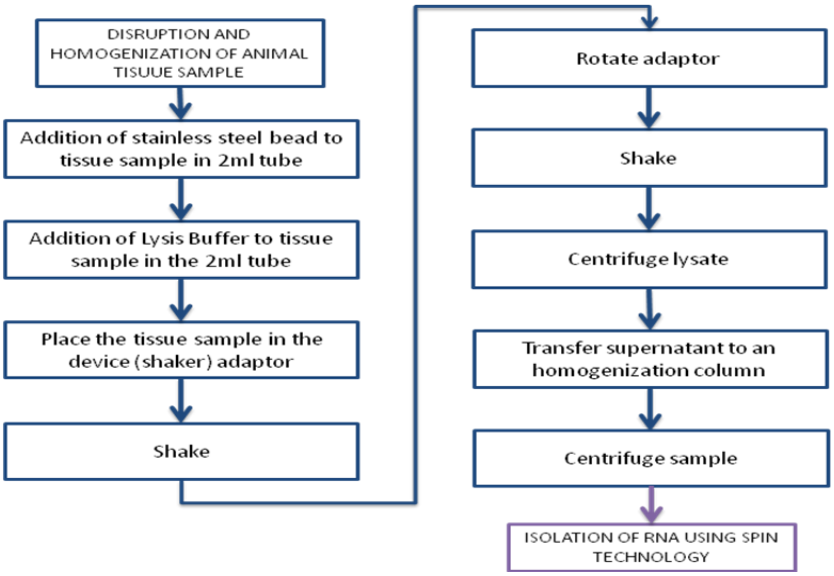

Figure 5. Disruption and Homogenization of Animal Tissue

Disruption and homogenization methods for animal tissue consist of mechanical disruption by using stainless steel beads and high frequency shaking (Figure 5).

First, a stainless steel bead is added to a 2ml tube containing the tissue sample. Second, the lysis buffer is added to the 2ml tube. Third, the sample is placed into an appropriate adaptor for the shaker. Fourth, the tissue sample is shaken to disrupt it. Fifth, the device adaptor is rotated to ensure that the complete sample is disrupted and homogenized. Sixth, the sample is shaken for a second period of time. Seventh, the tube is taken out of the shaker adaptor and placed into a centrifuge to separate the solid phase from the liquid phase. Eighth, the supernatant

is transferred to a homogenization column and finally spun to get the desired homogenized sample. The 2ml collection tube is then used for RNA isolation. For isolation of RNA from animal tissue the same procedure described for isolation of RNA from animal cells is followed (Figure 4).

Purification of Total RNA from Plant Cells

It is assumed that the plant sample has been previously properly harvested, stabilized, and store, and it is ready for RNA purification without any other pre-processing.

Disruption Method

Plant cell disruption: adding lysis buffer and votexing until the sample is lysed (Figure 6).

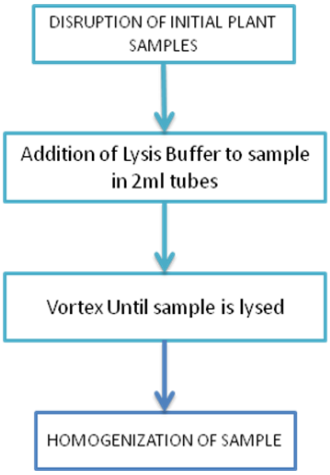

Figure 6. Disruption method for plant cells

Homogenization Method

For the homogenization method after transferring the lysate into the homogenization column and centrifuging the sample for a given period of time, the supernatant in the flow-through is transferred to a new 2ml collection tube and this supernatant is used for further RNA Isolation.

Isolation of RNA from plant cells is performed following the same procedure previously described for isolation of RNA from animal cells (Figure 4).

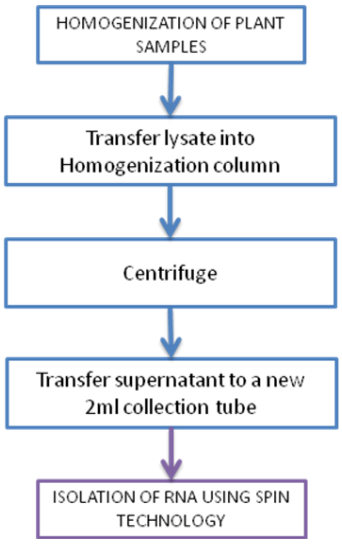

Figure 7. Homogenization method for plant cells: homogenization column and centrifugation

Purification of Total RNA Summary

Table 1 shows a summary of the steps used for the three RNA purification protocols described in this section.

Table 1. Protocols used for Total RNA Purification for different starting materials

<table>
<tr><th></th><th>Animal Cells</th><th>Plant Cells</th><th>Animal Tissue</th></tr>
<tr><td rowspan="2">Disruption of initial sample</td><td colspan="2">Addition of lysis buffer to cells in 2ml tubes Note: Different lysis buffers are used for different cells</td><td>Addition of stainless steel bead to tissue sample in 2ml tube</td></tr>
<tr><td colspan="2">Vortex until cells are lysed</td><td>Addition of lysis buffer to tissue sample in the 2ml tube</td></tr>
<tr><td rowspan="7">Homogenization of sample</td><td rowspan="2">Transfer lysate into homogenization column</td><td rowspan="2">Transfer lysate into homogenization column</td><td>Place the tissue sample in the device (shaker) adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td rowspan="3"></td><td rowspan="3">Centrifuge</td><td>Rotate adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td>Centrifuge lysate</td></tr>
<tr><td rowspan="2">Centrifuge</td><td rowspan="2">Transfer supernatant to a new 2ml collection tube</td><td>Transfer supernatant to an homogenization column</td></tr>
<tr><td>Centrifuge sample</td></tr>
<tr><td rowspan="11">Isolation of RNA using spin technology</td><td colspan="3">Addition of ethanol to the homogenized lysate</td></tr>
<tr><td colspan="3">Transfer the sample to a spin column placed in a 2ml tube</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge</td></tr>
</table>

Purification of genomic DNA

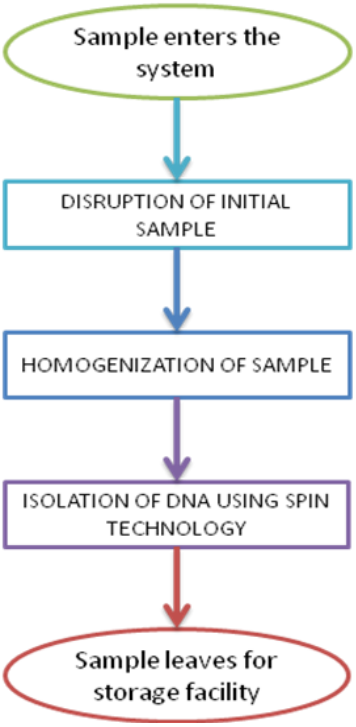

Figure 8. Process flow chart for Purification of Genomic DNA using spin technology

The complete protocol for purification of genomic DNA is divided into three main processes: disruption of the initial sample, homogenization of the sample, and isolation of DNA (Figure 8). First, biological samples are lysed and homogenized in the presence of a highly denaturing buffer, which immediately deactivates DNases to guarantee purification of undamaged DNA. Then, as DNA becomes bound to the chromatography column membrane, genomic DNA can be washed and eluted from the membrane.

Purification of Genomic DNA from Animal Cells

Disruption Method

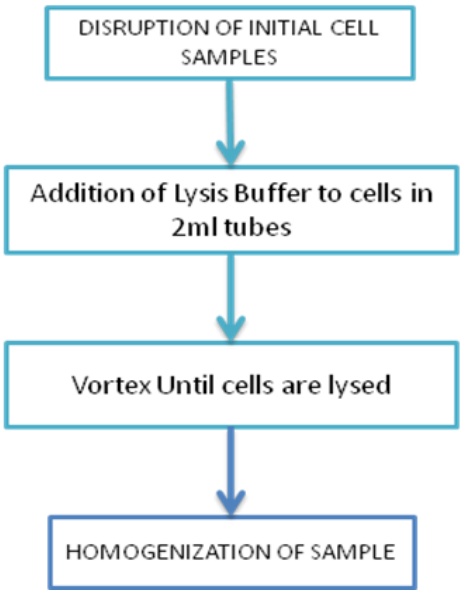

Figure 9. Disruption method for Animal Cells: Lysis Buffer and Vortex

Cells membranes are lysed by adding a Lysis Buffer to the sample contained in a 2ml tube and by vortexing the tubes (Figure 9). At this point the sample in the test goes from clear to milky. After the cells are lysed the sample proceeds to the homogenization process.

Homogenization Method

The homogenization method for animal cells consists on transferring the lysate to a homogenization column by pipetting the sample out of the 2ml collection tube and into the homogenization column. This is followed by the attaching of the column to a 2ml collection tube. The final homogenization step involves the placing of the tubes (columns and collection tubes) into a centrifuge, and spinning for a certain period of time to force the sample through the column's membrane (Figure 10).

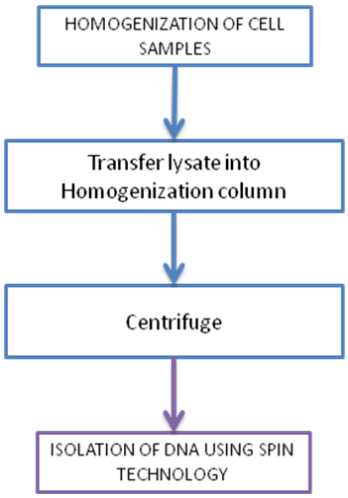

Figure 10. Homogenization method Animal Cells: Homogenization column and spin technology

Isolation of DNA Using Spin Technology

First, a wash buffer (called Wash Buffer) is added to the homogenization column where the genomic DNA is bound, the column is sealed and attached to a collection tube. Second, the sample is centrifuged for a given amount of time at a given temperature and the flow-through is discarded. Steps One and Two are repeated with a different wash buffer to ensure that all the undesirable particles are washed of the membrane leaving only genomic DNA bound to it before starting any elution steps. This second run of washing steps takes longer and is very important since the purpose is to dry the membrane from any residual trances of salt or ethanol that may interfere with future downstream reactions. Third, elution buffer is added to the sample, the column is sealed and attached to a new collection/elution tubes. Fourth, the sample is incubated for a given period of time at a given temperature. Fifth, the sample is centrifuged for a given period of time at a given temperature to elute the DNA. Repeat from step 3 to step 5 for complete recovery of DNA. The repetition of the last three steps is carried out to ensure complete recovery of DNA but at a lower concentration since it would have the complete DNA material but in twice the volume of elution buffer. In the chemical and biological environment solutions can be dissolved but the reverse process is not possible. If the sample requires a higher concentration, complete recovery of DNA is not possible. A decision has to be made between higher concentration and higher yield, both are not possible (Figure 11).

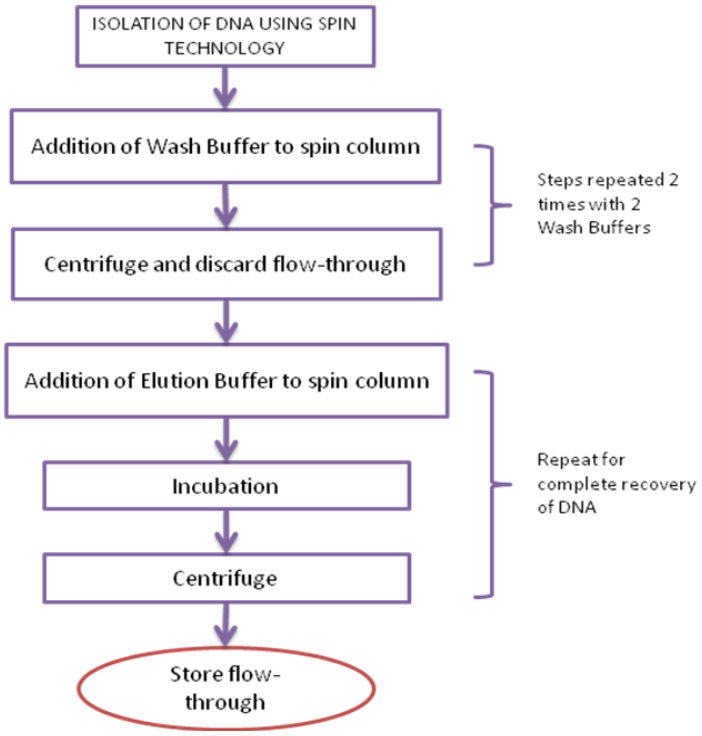

Figure 11. Isolation of Genomic DNA

Purification of Gnomic DNA from Animal Tissue

The starting material for the protocol has been previously harvested, stabilized, and stored. It is ready for DNA purification without any other pre-processing.

Disruption and Homogenization Methods

First, stainless steel beads are added to the tissue sample that is in a 2ml tube. Second, a given amount of lysis buffer if added to the tissue sample in the 2ml tube. Third, place the tissue sample in the shaker device adaptor. Forth, shake the sample for a given period of time. Fifth, rotate the adaptor before a second shake run so that the tissue sample can be completely disrupted. Sixth, shake the sample for a given period of time. Seventh, centrifuge the lysate to separate DNA from undesirable large particles. Eighth, transfer the supernatant to a homogenization column. Ninth, centrifuge the sample for a given period of time and discard flow-through. The homogenization column is then used for DNA Isolation as the genomic DNA is bound to the membrane of the column (Figure 12).

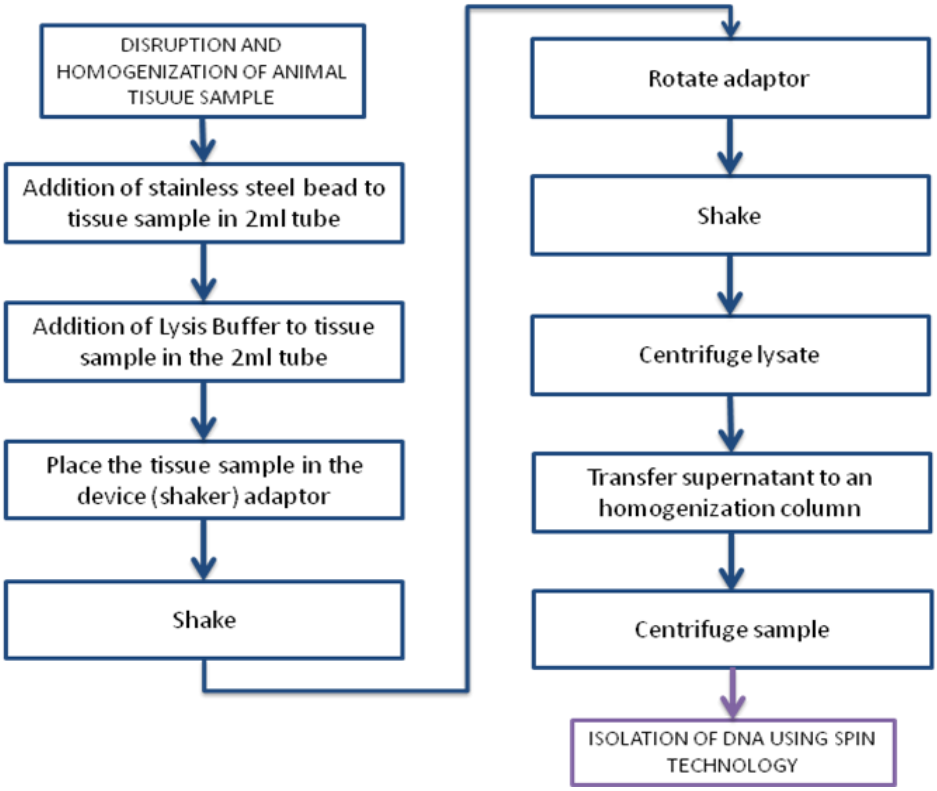

Figure 12. Disruption and Homogenization of Animal Tissue

For the actual isolation of DNA from animal tissue follow the same steps described for isolation of DNA from animal cells in the previous section (Figure 11)

Purification of Genomic DNA from Plant Cells

The starting material for the protocol has been previously harvested, stabilized, and stored. It is ready for DNA purification without any other pre-processing.

Disruption Method

The procedure for plant cell disruption is: add lysis buffer and votex until the sample is lysed (Figure 13)

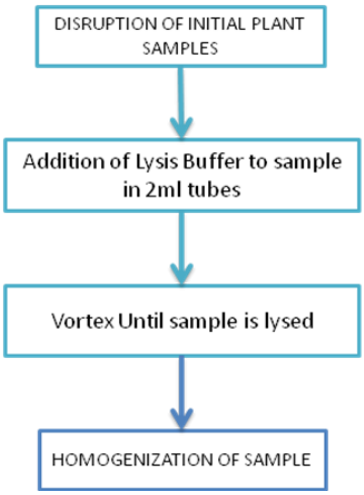

Figure 13. Disruption method for plant cells

Homogenization Method

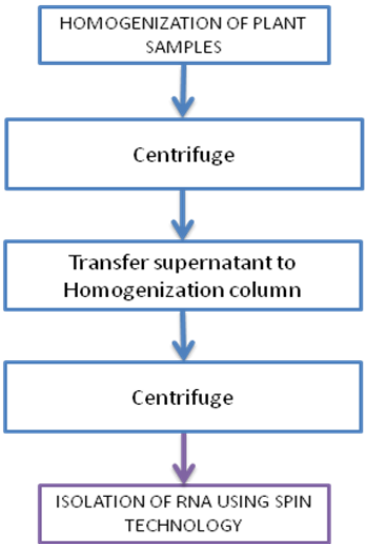

Figure 14. Homogenization method for plant cells

For the homogenization method the lysate is centrifuged for given period of time before the sample is transfer into the homogenization column. Then, the supernatant is transferred into the homogenization column and a 2ml collection tube is attached to the column. Afterwards, the sample is centrifuged for a given period of time at a given temperature and the flow-through is discarded and the homogenization column is used for further DNA isolation (Figure 14).

The Isolation method for DNA in this case is the one previously describe for isolation of genomic DNA using spin technology (Figure 11).

Purification of Genomic DNA Summary

Table 2 shows the protocol for Genomic DNA Purification from animal cells, animal tissue, and plant cells.

Table 2. Protocols used for Genomic DNA Purification for different starting materials

<table>
<tr><th></th><th>Animal Cells</th><th>Plant Cells</th><th>Animal Tissue</th></tr>
<tr><td rowspan="2">Disruption of initial sample</td><td colspan="2">Addition of lysis buffer to cells in 2ml tubes</td><td>Addition of stainless steel bead to tissue sample in 2ml tube</td></tr>
<tr><td colspan="2">Vortex until cells are lysed</td><td>Addition of lysis buffer to tissue sample in the 2ml tube</td></tr>
<tr><td rowspan="7">Homogenization of sample</td><td rowspan="3">Transfer lysate into homogenization column</td><td rowspan="2">Centrifuge</td><td>Place the tissue sample in the device (shaker) adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td rowspan="3">Transfer supernatant into homogenization column</td><td>Rotate adaptor</td></tr>
<tr><td>Shake</td></tr>
<tr><td rowspan="2">Centrifuge</td><td>Centrifuge lysate</td></tr>
<tr><td rowspan="2">Centrifuge</td><td>Transfer supernatant to an homogenization column</td></tr>
<tr><td>Centrifuge sample</td></tr>
<tr><td rowspan="10">Isolation of DNA using spin technology</td><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of wash buffer to spin column</td></tr>
<tr><td colspan="3">Centrifuge and discard flow-through</td></tr>
<tr><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Incubation</td></tr>
<tr><td colspan="3">Centrifuge</td></tr>
<tr><td colspan="3">Addition of elution buffer to spin column</td></tr>
<tr><td colspan="3">Incubation</td></tr>
<tr><td colspan="3">Centrifuge</td></tr>
</table>

APPENDIX B

DESIGN NOTES FOR THE MATLAB® SIMULATOR OF A SEQUENTIAL AUTOMATION LINE

The simulator developed to validate the queue clearing strategy and equipment parallelization strategy, describes the time behavior of any sequential automation line. The idea behind the simulator conception was to mimic the transfer of samples among the system station by basic operations and to rely on none of the formulas developed in Chapter 7. We simulate how the different samples flow through the system at any desired time. The simulator takes the parameters that describe the system. It then adheres to the input rate and the service times of each station and keeps count of all samples that go in and out of each station and the times of such arrival and departure events. The queue sizes and queue times for each of the stations and the number of parts in the processing unit of each station all become known at any given time.

The given system parameters that need to be specified by the program user are the number of stations, the parts' arrival period to the system, the number of parts arriving into the system at the same time, the service time for each of the stations, the number of parts that a station processes at the same time, the number of equipment or process units that run each station, the overall automation line shift time, and the desired throughput.

Names of the Input Variables:

- num_stations: Number of stations in the protocol to be simulated
- AP: Arrival period to the system (related to the input rate of the first station)
- NP: Number of parts that arrive simultaneously at the system (i.e. to the first station)
- tshift: The final time for the simulation which represents the length of time that the shift lasts
- TP: Desired throughput
- St: A vector of num_station elements, where each element represents the service time for each station

- Np: A vector of num_station elements, where each element represents the number of parts (samples) each station can process or has to process at the same time.
- Npu: A vector of num_station elements, where each element represents the number of parallel units serving each station

Definitions of Computed Variables:

- tb is the time unit that is used for the simulation to allow the user to capture every event happening during the time interval from 0 to tshift.
- tb_temp is the time vector used for plotting the various figures.
- Nin is an array used to keep track of the number of samples that are coming in or are in each station at any given time
- Qs is an array used to keep track of the number of samples that are in the queue of each a station at any given time
- Qt is an array used to keep track of the time the samples have waited in the queue of each station at any given time
- Nspu is an array used to keep track of the number of samples that are in the processing unit of each station at any given time
- Nout is an array used to keep track of the number of samples that are coming out or that had come out each station at any given time
- flag_ON is a vector of num_stations length. Each element represents the flag for each station. flag_ON: 1 -> ON: Nin(t,p) > 0 || Nout(t,p) < TP, or 0 -> OFF: Nin(t,p) == 0 || Nout(t,p) >= TP for station p
- flag_TP: Binary flag which indicates the total amount of the throughput that has entered the system to the first station. flag_TP: 1 -> Nin(t,1) == TP, or 0 -> Nin(t,1) <= TP
- flag_IDLE is a vector of num_stations length. Each element represents the flag for each station. An idle station is a station that has no queues and no parts

inside the processing unit. The station is idle until a part reaches the station to change its status. flag_IDLE: 1 -> station is idle, or 0 -> station is not idle

- flag_WAITING is a vector of num_stations length. Each element represents the flag for each station. A station is waiting if the station serves more than one part at a time and has not yet reached the number of parts that it can process to start running. flag_WAITING : 1 -> station is waiting, or 0 -> station is not waiting
- flag_BUSY is a vector of num_stations length. Each element represents the flag for each station. A busy station is processing samples. flag_BUSY: 1 -> station is busy, or 0 -> station is not busy
- ton is a vector of num_stations elements, and each element represents the time when the first part or group of parts start to get processed by each station during a shift.
- tfirst is a vector of num_stations elements, and each element represents the time when the first part arrives at each station during a shift.
- NP_temp is used whenever we do not wish to overwrite the original value of NP. NP is copied to NP_temp which at the end might be used for the last run of the stations in the shut down process
- Nmiss indicates the number of parts that have not yet gone into the system
- Ap_temp is a vector of num_stations elements and each element represents the arrival period to each station
- St_temp is used in order not to overwrite the original value of St.
- Npp is a vector that represents the number of parts that a station can actually process taken into consideration parallel units.
- TP_station shows when the TP is reached in each of the stations.
- clock is a vector that keeps track how long a station has been servicing a part or group of parts.
- Nini indicates the instantaneous number of parts going into each of the stations

- Noutii indicates the instantaneous number of parts going out of each of the stations
- flag_IN indicates if a part or a group of parts has arrived at a station (1) or not (0)
- flag_OUT indicates if a part or a group of parts is leaving the station (1) or not(0)
- k is the time index for all vectors

Simulation Events:

First, the simulator checks if a sample is arriving at a station:

For Station 1: whenever the time is a multiple of the system's arrival period and the full expected throughput have not yet been attained, a part enters the station.

For the remaining stations: If a part comes out of the previous station at time t, it will enter the next station instantly. The assumption is that the inter-station transportation time is zero.

Thereafter, the simulator checks if each of the stations has finished serving the sample and if a part is coming out of the station. In this case, for all stations, if a station has already run its nominal service time, it is assumed that it has finished servicing the part(s) inside its processing unit.

Then the program calculates and updates Nin(t,p):

If it is detected that a new part or group of parts went into the processing unit at time t, that number of parts is added to the total number of parts that have gone into the station. For the first station it would be the number of parts that enter the system at a time and for the rest of the stations it is the number of parts coming from the prior stations.

After the program finds if a part is coming in or out of a given station and if the array variables have been updated, depending on the status of the stations, IDLE, WAITING or BUSY, the simulator then calculates the desired outputs in the following way.

If the status of a station is IDLE:

If a station is idle the only other event that may have occurred is that a part arrives at the station (Flag_IN). If the station is OFF this will set the times tfirst and ton of the given stations depending on the number of parts entering the station: If it is less than Np of the given station it would only set tfirst, if is equal to or greater than Np it will also set ton and turn ON station. As

the station is IDLE nothing has came out of the station, and the current Nout equals to previous Nout and instant Nout is zero.

The consideration of the last part IN is done differently because the station has to be turned off after that last part is processed. If Nin is equal to the throughput that means that no more part(s) will arrive after t. In order to empty the station the BUSY flag is set so that the station starts processing the part(s) even if Nspu is smaller than Np.

If the part in question is not the last part IN of the given station, and if the number of parts that come in from a prior station is less than Np, the parts will go to the processing unit and wait for more parts to arrive. On the other hand, if the number of parts that are coming in is equal to Np, the station starts to process them. A third case will be if the number of parts arriving at the station is greater than Np. In this last case Np parts will go to the process units to start being processed and the rest will go and wait in the queue.

If the status of the station is IDLE and no parts are coming in, the station will continue in its idle state.

If the status of the station is WAITING:

If a station is waiting then in this case also the only other event that can occur at the same time is that a part arrives at the station (Flag_IN). If after time t the station is ready to start processing parts, ton is set and the station will be turned ON. As nothing could come out of the station, the current Nout equals to the previous Nout and the instant Nout is zero.

The last part IN case has to be taken into consideration separately again. If Nin is equal to throughput that means that no more part(s) will arrive after t. In order to empty the station the BUSY flag is set so that the station starts processing the part(s) even if Nspu is smaller than Np. If it is not the last part IN and the number of parts that comes in from a prior station is less than the empty spaces in the process unit, the parts will go to the process unit and wait for more parts to arrive. If the number of parts coming into the station is equal to the empty spaces in the processing unit, the station starts to process them. Last, if the number of parts arriving at the station is greater than the number of empty spaces in the processing unit, Np parts will go to the process units to start being processed and the rest will go and wait in the queue.

If no part(s) is arriving to the station the station will remain in WAITING status.

If the status of the station is BUSY:

There are four different cases to consider if a station is in a busy state: one, if nothing is coming in or going out of the station; two, if parts are coming into the station but nothing is coming out of the station; three, if nothing is coming into the station but something is coming out of the station; and four, if a part is coming into the station and at the same time another part is going out of the station.

If no event occurs (no parts are coming in or going out of the station) the stations clock is increased to keep track of the service time of the part that is currently being processed.

If a part is coming into the station but nothing is going out, the part(s) will be placed in the stations queue because as the status of the station is busy, a part is being processed at the time when the other part arrives.

Case three is whenever parts are coming out of the station and no parts are coming into the station. One has to consider if the part that is coming in is the last part of the shift or not. If the part that is coming in is the last part to be processed and if the part is the last part in the queue, the station is kept in BUSY state so that the station starts processing the part(s) even if Nspu is smaller than Np. If the queue for the given station is larger than Np, Np parts will start to be served. If the queue is shorter than Np all parts will go into the unit, but it will change to WAITING status. The station will wait for more parts to arrive at a later time to start processing the parts. If the queue size is zero the station goes to IDLE status.

If parts are coming IN and going OUT the parts will be treated in a manner that combines both of the previous cases. If Nin is equal to the throughput that means that no more part(s) will arrive after t. In order to empty the station the BUSY flag is set so that the station starts processing the part(s) even if Nspu is smaller than Np. If it is not the last part going IN and the number of parts going into the station plus the queue size is smaller than Np the station will go into WAITING status. If parts are going into the station plus the queue size equals to Np all parts will go into the processing unit and start to be served. If parts coming in plus the queue size is

larger than Np the remaining parts will be placed in the queue and the others will start to be served in the process unit.

If the last group to be served is not a multiple of NP, NP value is changed to be the number of remaining parts to allow that all required parts are served.

If TP has already gone in to the first station the shut down process will start.

Once the output arrays are calculated, these are plotted to illustrate the behavior of the system throughout shift period.

For the parallelizing strategy the simulator is used to evaluate each of the possible scenarios. By running the simulator and exploring the results found, with the help of the critical station equations (and this is the only time that Chapter 7 equations are used), a parallelizing decision is taken. The parameters are changed to represent the new system and it is simulated one more time. The new results are explored and evaluated using the critical station criteria. The process will terminate once the resulting system follows all restrictions that the protocol has.

The simulator was developed as an m-file function. For this reason, input variables need to be initialized before the function can be called and used:

function [Nin, Qs, Qt, Nspu, Nout, ton, tb_temp, tfirst, Ap_temp, TP_station] =

simulator_para_sync (**num_stations, AP, NP, tshift, TP, St, Np, Npu, t_stop, t_resume**)

APPENDIX C

SIMULATOR CODE – BASIC PART TRANSFER BETWEEN STATIONS

```
%-----------------------------------------------------------------------
---
% Simulator
%-----------------------------------------------------------------------
---

function [Nin, Qs, Qt, Nspu, Nout, ton, tb_temp, tfirst, Ap_temp,
TP_station] = ...
    simulator_para_sync (num_stations, AP, NP, tshift, TP, St, Np, Npu,
t_stop, t_resume)
%-----------------------------------------------------------------------
---
%Input Variables:
%num_stations: Number of stations in the protocol thta will be
simulated
%AP: Arrival period to the system (first station)
%NP: Number of parts that arrive at the same time to the system (first
station)
%tshift: The lnght of time the shift lasts and therefore the simulation
will be executed for this period of time
%TP: Desired tthroughput
%St: A vector of num_station elements, and each element represents the
service time for each station
%Np: A vector of num_station elements, and each element represents the
number of parts (samples) a station can process or has to process at
the same time.
%Npu: A vector of num_station elements, and each element represents the
number of parallel units serving ecah station
%-----------------------------------------------------------------------
---
%Definitions
%-----------------------------------------------------------------------
---
tb = aura_time_base(AP, St);%The time unit that is used for the
simulation to allow us capture every event in tshift
tb_temp = zeros(1,floor(tshift/tb) + 1);
%tb_temp is the time vector used for ploting figures.

Nin = zeros(floor(tshift/tb) + 1,num_stations);%Array used to keep
track of the number of samples that are coming in or are in the station
at any given time
Qs = zeros(floor(tshift/tb) + 1,num_stations);%Array used to keep track
of the number of samples that are in the queue of a station at any
given time
Qt = zeros(floor(tshift/tb) + 1,num_stations);
Nspu = zeros(floor(tshift/tb) + 1,num_stations);%Array used to keep
track of the number of samples that are in the process unit of a
station at any given time
Nout = zeros(floor(tshift/tb) + 1,num_stations);%Array used to keep
track of the number of samples that are coming out or that had come out
the station at any given time
```

```
flag_ON = zeros(1,num_stations);%Is a vector of num_stations lenght.
Each element represents the flag for each station.
%flag_ON indicates if station p: 1 -> ON: Nin(t,p) > 0 || Nout(t,p) <
TP
%                                 0 -> OFF: Nin(t,p) == 0 || Nout(t,p)
>= TP
flag_TP = 0;%Binary flag which indicates is the total amount of the
throughput has enetered the system (first station)
%flag_TP indicates if : 1 -> Nin(t,1) == TP
%                       0 -> Nin(t,1) <= TP
flag_IDLE = ones(1,num_stations);%Is a vector of num_stations lenght.
Each element represents the flag for each station.
%An idle station is an station that has no queues and nothing in the
process unit to process (serve). The station is idle until a parts
reaches the station to change its status.
%flag_IDLE indicates if : 1 -> station is idle
%                         0 -> station is not idle
flag_WAITING = zeros(1,num_stations);%Is a vector of num_stations
lenght. Each element represents the flag for each station.
%An station is waiting if the station serves more than one part at a
time and hasn't reach the number of parts that it can process to start
running.
%flag_WAITING indicates if : 1 -> station is waiting
%                            0 -> station is not waiting
flag_BUSY = zeros(1,num_stations);%Is a vector of num_stations lenght.
Each element represents the flag for each station.
%A busy station is processing samples.
%flag_BUSY indicates if : 1 -> station is busy
%                         0 -> station is not busy

ton = zeros(1,num_stations); %ton is a vector of num_stations elements,
and each element represents the time when the first part or group of
parts start to get processes by a station in a shift.
tfirst = zeros(1,num_stations); %tfirst is a vector of num_stations
elements, and each element represent the time whent the first part
arrives to a station in a shift.

NP_temp = NP;%Used to not to overwrite the original value of NP, NP is
copied to NP_temp which at the end might be used for the last run of
the stations in the shut down process
Nmiss = TP;%Number of parts that hasn't yet gone into the system
Ap_temp = zeros(1,num_stations);%A vector of num_stations elements and
each element represent the arrival period to each station
St_temp = St;%Used to not to overwrite the original value
Npp = Np.*Npu;%Is a vector that represent the number of parts that a
station can actualy process taken in to consideration parallel units.
TP_station = zeros(1,num_stations);%shows when the TP (Nout(p) = TP) is
achived in each of the stations (index number, not time value).

Ap_temp(1) = AP;
for i = 2:num_stations
    Ap_temp(i) = max(Ap_temp(i - 1),St_temp(i-1));
end

clock = zeros(1, num_stations);%A vector that keeps track how long a
station has been servicing a part or gruop of parts.
```

```
Nini = zeros(1, num_stations);%Instant number of parts in to each of
the stations
Nouti = zeros(1, num_stations);%Instant number of parts out to each of
the stations

flag_IN = 0;%Indicates if a part(s) has arrived to a station (1) or not
(0)
flag_OUT = 0;%Indicates if a part(s) is leaving the station (1) or
not(0)

%-----------------------------------------------------------------------
---
%Clearing Strategy
%-----------------------------------------------------------------------
---
flag_originalsys = 1;
if t_stop < tshift
    flag_originalsys = 0;
end
t_temp = 0;
%-----------------------------------------------------------------------
---

%-----------------------------------------------------------------------
---
%Initialization
%-----------------------------------------------------------------------
---
k = 0; %k is the time index for all vectors
%-----------------------------------------------------------------------
---

for t = 0:tb:tshift
    k = k + 1;
    tb_temp(k) = t;
    for p = 1:num_stations
        flag_IN = 0;
        flag_OUT = 0;
        Nini(p) = 0;
        %Check if a sample is arrivng to a station:
        %   Station 1: whenever the time is a multiple of the system's
        %    arrival period and that not all the troughput have been
served
        %   a part enters the station.
        if p == 1 && flag_TP == 0
            if mod(t,AP) == 0 && flag_originalsys == 1
                Nini(p) = NP_temp;
                flag_IN = 1;
            elseif  flag_originalsys  ==  0  &&  t_temp  <=  t_stop  &&
mod(t_temp,AP) == 0
                Nini(p) = NP_temp;
                flag_IN = 1;
            end
        %    Remaining Stations: If a part comes out of the station at
time
        %   t, it will enter the next station.
```

```
        elseif p > 1 && Nouti(p - 1) > 0
            Nini(p) = Nouti(p - 1);
            flag_IN = 1;
            Nouti(p - 1) = 0;
        end
        %Check if the station finished serving the samples
        %   For all stations: If a station has already ran for St
        %    it has finished servicing the part(s) inside the process
unit.
        if clock(p) == St(p)
            Nouti(p) = Npp(p);
            flag_OUT = 1;
        end
        %Calculates new current Nin(t,p)
        %   If it was detected that a new part(s) went into the process
        %   unit at time t, that number of parts is added to the total
        %   number of parts that have gone into de station. For station
it
        %    would be the number of parts that enter the system at a
time
        %    and for the rest of the stations is the number of parts
coming
        %   from the prior station.
        if flag_IN == 1 && k == 1
            Nin(k,p) = NP;
        elseif k > 1
            Nin(k,p) = Nin(k - 1,p) + Nini(p);
        end
%-----------------------------------------------------------------------
---
%-----------------------------------------------------------------------
---
        %Status statio p: IDLE
        %   If a station is idle the only other event that can ocurr is
        %     that a part arrives to the station (Flag_IN). If the
station is
        %   OFF this will set the times tfirst and ton of the
        %    given stations depending on the number of parts entering
the
        %   station: If is less than Np of the given station it would
only
        %   set tfirst, if is equal to or greater than Np will also set
        %   ton and turn On station.
        if flag_IDLE(p) == 1 && flag_IN == 1
            if flag_ON(p) == 0
                tfirst(p) = t;
                if Nin(k,p) >= Npp(p);
                    flag_ON(p) = 1;
                    ton(p) = t;
                end
            end
            %As nothing has came out of the station, current Nout
equals to
            %previous Nout and instant Nout is zero.
            if k > 1
                Nout(k,p) = Nout(k - 1,p);
                Nouti(p) = 0;
```

```
            end
            %Last part IN
            %   If Nin is equal to throughput that means that no more
            %    part(s) will arrive after t, in order to empty the
station
            %     the BUSY flag is set so that the station starts
processing
            %   the part(s) even if Nspu is smaller than Np.
            if Nin(k,p) == TP && Nin(k,p) - Nout(k,p) < Npp(p)
                Nspu(k,p) = Nini(p);
                flag_IDLE(p) = 0;
                flag_BUSY(p) = 1;
                clock(p) = tb;
            %If not last part IN
            %    If the number of parts that comes in from a prior
station
            %   is less than Np, the parts will go to the process unit
and
            %   wait for more parts to arrive.
            elseif Nini(p) < Npp(p)
                Nspu(k,p) = Nini(p);
                flag_IDLE(p) = 0;
                flag_WAITING(p) = 1;
            %   If the number of parts coming in are equal to Np, the
            %   station starts to process them.
            elseif Nini(p) == Npp(p)
                Nspu(k,p) = Nini(p);
                flag_IDLE(p) = 0;
                flag_BUSY(p) = 1;
                clock(p) = tb;
            %   If the number of parts arriving if greater than Np, Np
            %    parts will go to the process units to start being
processed
            %   and the rest will go and wait in the queue.
            elseif Nini(p) > Npp(p)
                Nspu(k,p) = Npp(p);
                Qs(k,p) = Nini(p) - Npp(p);
                Qt(k,p) = ceil(Qs(k,p)/Npp(p))*St(p);
                flag_IDLE(p) = 0;
                flag_BUSY(p) = 1;
                clock(p) = tb;
            end
        % If no part(s) is arriving to the station the station will
remain in
        % an IDLE status.
        elseif flag_IDLE(p) == 1 && flag_IN == 0 && k > 1
            Nout(k,p) = Nout(k - 1,p);
%-----------------------------------------------------------------------
---
        %Status station p: WAITING
        %   If a station is waiting the only other event that can ocurr
is
        %   that a part arrives to the station (Flag_IN). If after time
t
        %   the station is ready to start procesing parts, ton will be
set
        %   and the station will be turned ON.
```

```
        elseif flag_WAITING(p) == 1 && flag_IN == 1
            if flag_ON(p) == 0 && Nin(k,p) >= Npp(p);
                flag_ON(p) = 1;
                ton(p) = t;
            end
            %As  nothing  has  came  out  of  the  station,  current  Nout
equals to
            %previous Nout and instant Nout is zero.
            if k > 1
                Nout(k,p) = Nout(k - 1,p);
                Nouti(p) = 0;
            end
            %Last part IN
            %   If Nin is equal to throughput that means that no more
            %    part(s)  will  arrive  after  t,  in  order  to  empty  the
station
            %     the  BUSY  flag  is  set  so  that  the  station  starts
processing
            %   the part(s) even if Nspu is smaller than Np.
            if Nin(k,p) == TP && Nin(k,p) - Nout(k,p) < Npp(p)
                Nspu(k,p) = Nspu(k - 1,p) + Nini(p);
                flag_WAITING(p) = 0;
                flag_BUSY(p) = 1;
                clock(p) = tb;
            %If not last part IN
            %    If  the  number  of  parts  that  comes  in  from  a  prior
station
            %   is less than the empty paces in the process unit, the
parts
            %   will go to the process unit and wait for more parts to
            %   arrive.
            elseif Nini(p) < Npp(p) - Nspu(k - 1,p)
                Nspu(k,p) = Nspu(k - 1,p) + Nini(p);
            %   If the number of parts coming in are equal to the empty
            %    spaces  in  the  process  unit,  the  station  starts  to
process
            %   them.
            elseif Nini(p) == Npp(p) - Nspu(k - 1,p)
                Nspu(k,p) = Npp(p);
                flag_WAITING(p) = 0;
                flag_BUSY(p) = 1;
                clock(p) = tb;
            %    If  the  number  of  parts  arriving  if  greater  than  the
number
            %   of empty spaces in the process unit, Np parts will go
to
            %   the process units to start being processed and the rest
            %   will go and wait in the queue.
            elseif Nini(p) > Npp(p) - Nspu(k - 1,p)
                Nspu(k,p) = Npp(p);
                Qs(k,p) = Nini(p) - (Npp(p) - Nspu(k - 1,p));
                Qt(k,p) = ceil(Qs(k,p)/Npp(p))*St(p);
                flag_WAITING(p) = 0;
                flag_BUSY(p) = 1;
                clock(p) = tb;
            end
```

```
        % If no part(s) is arriving to the station the station will
remain in
        % an WAITING status.
        elseif flag_WAITING(p) == 1 && flag_IN == 0 && k > 1
            Nout(k,p) = Nout(k - 1,p);
            Nspu(k,p) = Nspu(k - 1,p);
%-----------------------------------------------------------------------
---
        %Status station p: BUSY
        elseif flag_BUSY(p) == 1
            if k > 1
                Nout(k,p) = Nout(k - 1,p);
                Nouti(p) = 0;
            end
            %If no event occurs the stations clock is increased
            if flag_IN == 0 && flag_OUT == 0
                clock(p) = clock(p) + tb;
                Nspu(k,p) = Nspu(k - 1,p);
                Qs(k,p) = Qs(k - 1,p);
                if Qt(k - 1,p) > 0
                    Qt(k,p) = Qt(k - 1,p) - tb;
                end
            %If a part is comming into the station but nothing is going
            %out, the part(s) will be placed in the stations queue.
            elseif flag_IN == 1 && flag_OUT == 0
                Qs(k,p) = Qs(k - 1,p) + Nini(p);
                Qt(k,p) = Qt(k - 1,p) + (St(p) - clock(p));
                Nspu(k,p) = Nspu(k - 1,p);
                clock(p) = clock(p) + tb;
            %If parts are coming out and no parts coming in.
            elseif flag_IN == 0 && flag_OUT == 1
                Nouti(p) = Nspu(k - 1,p);
                Nout(k,p) = Nout (k - 1,p) + Nouti(p);
                %Last part to be processed
                %   if the part is the last part in the queue in order
to
                %   empty the station the BUSY flag is set so that the
                %   station starts processing the part(s) even if Nspu
is
                %   smaller than Np.
                if Nin(k,p) == TP && Nin(k,p) - Nout(k,p) < Npp(p)
                    Nspu(k,p) = Qs(k - 1,p);
                    clock(p) = tb;
                %   If queue is larger than Np, Np parts will start to
be
                %   served.
                elseif Qs(k - 1,p) >= Npp(p)
                    Nspu(k,p) = Npp(p);
                    Qs(k,p) = Qs(k - 1,p) - Npp(p);
                    Qt(k,p) = Qt(k - 1,p) - tb;
                    if Qs(k,p) == 0
                        Qt(k,p) = 0;
                    end
                    clock(p) = tb;
                %   If queue is shorter than Np all parts will go into
the
                %   unit, but it will change to WAITING status.
```

```
                elseif Qs(k - 1,p) > 0 && Qs(k - 1,p) < Npp(p)
                    Nspu(k,p) = Qs(k - 1,p);
                    Qs(k,p) = 0;
                    Qt(k,p) = 0;
                    clock(p) = 0;
                    flag_BUSY(p) = 0;
                    flag_WAITING(p) = 1;
                %    If queue size is zero the station goes to IDLE
status.
                elseif Qs(k - 1,p) == 0
                    Nspu(k,p) = 0;
                    Qs(k,p) = 0;
                    Qt(k,p) = 0;
                    clock(p) = 0;
                    flag_BUSY(p) = 0;
                    flag_IDLE(p) = 1;
                end
            %If parts are coming IN and going OUT
            elseif flag_IN == 1 && flag_OUT ==1
                Nouti(p) = Nspu(k - 1,p);
                Nout(k,p) = Nout (k - 1,p) + Nouti(p);
                %if last part IN:
                %    If Nin is equal to throughput that means that no
more
                %   part(s) will arrive after t, in order to empty the
station
                %    the BUSY flag is set so that the station starts
processing
                %   the part(s) even if Nspu is smaller than Np.
                if Nin(k,p) == TP && Nin(k,p) - Nout(k,p) < Npp(p)
                    Nspu(k,p) = Nini(p);
                    clock(p) = tb;
                %If not last part going IN:
                %    If the number of parts going into the station plus
the
                %    queue size is smaller than Np the station will go
into
                %   WAITING status.
                elseif Nini(p) + Qs(k - 1,p) < Npp(p)
                    Nspu(k,p) = Nini(p) + Qs(k - 1,p);
                    Qs(k,p) = 0;
                    Qt(k,p) = 0;
                    clock(p) = 0;
                    flag_BUSY(p) = 0;
                    flag_WAITING(p) = 1;
                %    If parts going into the station plus queue size
equals
                %   to Np all parts will go into de processing unit and
                %   start to be served.
                elseif Nini(p) + Qs(k - 1,p) == Npp(p)
                    Nspu(k,p) = Nini(p) + Qs(k - 1,p);
                    Qs(k,p) = 0;
                    Qt(k,p) = 0;
                    clock(p) = tb;
                %    If parts coming in plus queue size is larger than
Np
```

```
                %   the remaining parts will be placed in the queue and
the
                %   others will start to be served in the process unit.
                elseif Nini(p) + Qs(k - 1,p) > Npp(p)
                    Nspu(k,p) = Npp(p);
                    Qs(k,p) = Nini(p) + Qs(k - 1,p) - Npp(p);
                    Qt(k,p) = ceil(Qs(k,p)/Npp(p))*St(p);
                    clock(p) = tb;
                end
            end
        end
        if TP_station(p) == 0 && Nout(k,p) == TP
            TP_station(p) = k;
        end
    end
    %If the last gorup to be served is not a multiple of NP, this is
    %changed to allow that all required parts are served.
    if flag_TP == 0
        if flag_originalsys == 1 && mod(t,AP) == 0
            Nmiss = Nmiss - NP;
            if Nmiss < NP
                NP_temp = Nmiss;
            end
        elseif  flag_originalsys  ==  0  &&  t_temp  <=  t_stop  &&
mod(t_temp,AP) == 0
            Nmiss = Nmiss - NP;
            if Nmiss < NP
                NP_temp = Nmiss;
            end
        end
    end
    %If TP has already gone in to the first station the shut down
process
    %will start.
    if Nin(k,1) == TP
        flag_TP = 1;
    end
    t_temp = t_temp + tb;
    if t_temp == t_resume
        t_temp = 0;
    end
end

figure
subplot(2,1,1)
plot(tb_temp,Nin)
title('Nin(p)')
subplot(2,1,2)
plot(tb_temp,Nout)
title('Nout(p)')
figure
subplot(2,1,1)
plot(tb_temp,Qs)
title('Qs(p)')
subplot(2,1,2)
plot(tb_temp,Qt)
title('Qt(p)'
```

REFERENCES

Amersham Pharmacia Biotech. *The Recombinant Protein Handbook: Protein Amplification and Simple Purification*. Amersham Pharmacia Biotech, 2011

Askin, Ronald G., and Charles R. Standridge. *Modeling and Analysis of Manufacturing Systems*. New York: Wiley, 1993. 167.

Ausubel, Frederick M., Roger Brent, Robert E. Kingston, David D. Moore, J. G. Seidman, John A. Smith, and Kevin Struhl. *Short Protocols in Molecular Biology*. 5th ed. Vol. 1. New York,NY John Wiley & Sons, 2002.

Banks, M. N., Cacace, A. M., O'Conell, J. et al. "High-Throughput Screening: Evolution of Technologies and Methods." *Drug Discovery Handbook, Wiley*, 2007 p. 559 - 602.

Bartfai, T. & G. V. Lees. *Drug Discovery from bedside to Wall Street.* Elsevier Academic Press. 2006 Burlington, MA.

Beckman Coulter, *Labsystems MultiDrop 384*, Beckman Coulter, 2011, https://www.beckmancoulter.com/

Berthold Technologies. "Centro LB 960 Microplate Luminometer." Berthold Technologies. Accessed November 10, 2014. https://www.berthold.com/en/bio/microplate_luminometer_centro_LB960.

Cardona, Aura Maria, and Zvi Roth. "Design Considerations in High-Throughput Automation for Biotechnology." *Florida Conference on Recent Advances in Robotics, FCRAR 2008* Melbourne, FL (2008).

Cardona, Aura Maria, Chingping Han, and Zvi Roth. "Optimization of Multi-Centrifuge Steps in Biotechnology Automation." *Florida Conference on Recent Advances in Robotics, FCRAR 2009*, Jupiter, FL (2009).

(Cardona, 2012a) Cardona, Aura Maria, Zvi Roth, and Chingping Han. "High-Throughput Automation Design Considerations for Biotechnology Processes Involving RNA

Purification Protocols Using Multi-Centrifuge Bioseparation Steps." *Journal of Robotics and Computer Integrated Manufacturing* 28, no. 3 (2012): 285–293.

(Cardona, 2012b) Cardona, Aura Maria, and Zvi Roth. "Throughput Specifications for Upstream Biotechnology Process Automation to Meet the Needs of Automated Drug Discovery Campaigns." *Florida Conference on Recent Advances in Robotics, FCRAR 2012*, Boca Raton, FL (2012).

(Cardona, 2014a) Cardona, Aura Maria, Zvi Roth, and Chingping Han. "Modular Implementation of Laboratory Unit Operations (LUOs) for Automation of Biotechnology Protocols." *Florida Conference on Recent Advances in Robotics, FCRAR 2014*, Miami, FL (2014).

(Cardona, 2014b) Cardona, Aura Maria, Zvi Roth, and Chingping Han. "Group Technology (GT) Applied to Biotechnology Automation." *Florida Conference on Recent Advances in Robotics, FCRAR 2014*, Miami, FL (2014).

Carnero, Amancio. "High Throughput Screening in Drug Discovery." *Clinical and Translational Oncology* 8, no. 7 (2006): 482-90.

Chung, C.C., Ohwaki, K., Scheneeweis, J.E., et al. “A Fluorescence-Based Thiol Quantification Assay for Ultra-High-Throughput Screening for Inhibitors of Coenzyme A Production”. *ASSAY and Drug Development Technologies*, Volume 6, Number 3, 2008, p. 361-374, DOI: 10.1089/adt.2007.105.

CISBIO, *FlexStation 3*, CISBIO – Bioassays, 2011, http://www.htrf.com/splash.asp

Daoura, Marco J., and Deirdre Meldrum. "Precise Automated Control of Fluid Volumes Inside Glass Capillaries." *IEEE Journal of Microelectromechanical Systems* 8, no. 1 (1999): 71-77.

Drews, Jürgen. "Drug Discovery: A Historical Perspective." *Science* 287, no. 5460 (2000): 1960-964.

DYNEX, *Immulux™,* DYNEX, 2011, http://www.dynextechnologies.com/

Eckstein, J. *ISOA/ARF Drug Development Tutorial*. Institute for the Study of Aging, Alzheimer Research Forum. 2006

Eppendorf. "Eppendorf Centrifuge 5427 Refrigerated." POCD Scientific. January 1, 2013. Accessed November 10, 2014. http://www.pocdscientific.com.au/eppendorf_centrifuge_5427R.php.

FANUC. "FANUC M-430iA Robotic Food Case Packing System - Courtesy of Schneider Packaging." FANUC. Accessed November 10, 2014. http://robot.fanucamerica.com/robot-applications/m430ia-robotic-case-packing-system.aspx.

Florida Atlantic University. *Biotechnology Laboratory I Manual*. Boca Raton, FL, Biological Science, Florida Atlantic University, 2006.

Geisler, Thomas, Johann Ressler, Hartmann Harz, Bernhard Wolf, and Rainer Uhl. "Automated Multiparametric Platform for High-Content and High-Throughput Analytical Screening on Living Cells." *IEEE Transactions on Automation Science and Engineering* 3, no. 2 (2006): 169-76.

Girdhar, Anupam. *Expansion of Group Technology Part Coding Based on Functionality*. Ms Thesis, University of Cincinnati, 2001.

Groover, M. *Fundamentals of Modern Manufacturing Materials, Processes, and Systems*. Upper Saddle River, NJ: Prentice Hall, 1996.

Hayworth, Douglas, and Thermo Scientific. "Cell Lysis Solutions." Pierce Protein Biology. Accessed November 10, 2014. http://www.piercenet.com/method/cell-lysis-solutions.

Hodder, P., Baillargeon, P., Chase, P., DeLuca, L., Madoux, F., Mercer, B., Minond, D., Saldanha, S., Scampavia, L., Spicer, T., Subramaniam, P., and Sullivan, L. “Probe and Drug Discovery: The Lead Identification Department”. *Translational Research Institute, Scientific Report 2007*.

Investopedia. "Cost Of Capital Definition." Investopedia. Accessed November 10, 2014. http://www.investopedia.com/terms/c/costofcapital.asp.

Jaganathan, Jai Kannan Janaki. *Solution to Large Facility Layout Problems Using Group Technology*. Ms Thesis, Wichita State University, 2007.

Kanji, Siddharth, and Quora. "What Is OPITZ Classification System in Group Technology?" Quora. Accessed November 10, 2014. http://www.quora.com/What-is-OPITZ-classification-system-in-Group-Technology.

Kelton, W. David, Randall P. Sadowski, and Deborah A. Sadowski. *Simulation with Arena*. 2nd ed. New York, NY: McGraw-Hill, 2002.

Kimball. "Junctions Between Cells." Kimball's Biology Pages. April 27, 2014. Accessed November 10, 2014. http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/J/Junctions.html.

Meldrum, Deirdre R. "Requirements for Automated Sample Handling in Genome Analysis." *IEEE*, 1994, 1047, 1048.

Meldrum, Deirdre R. "Engineering in Genomics." *IEEE Engineering in Medicine and Biology*, 1995, 443-448.

Meldrum, Deirdre R. "A Biomechatronic Fluid Sample Handling System for DNA Processing." *IEEE/ASME Transactions on Mechatronics* 2, no. 2 (1997): 99-109.

Meldrum, Deirdre R., Harold T. Evensen, William H. Pence, Stephen E. Moody, David L. Cunningham, and Peter J. Wiktor. "ACAPELLA, A Capillary-Based Submicroliter Automated Sample Preparation System for Genome Analysis." *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* Atlanta, USA (1999).

(Meldrum, 2000a) Meldrum, D.R., and S.E. Moody. "Automating High-Throughput Fluid Sample Handling for Biotechnology and Chemestry." *Proceeding on the 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA (2000): 198-205.

(Meldrum, 2000b) Meldrum, Deirdre R., Harold T. Evensen, William H. Pence, and Stephen E. Moody. "ACAPELLA-1K: A Biomechatronic Fluid Handling System for Genome Analysis That Processes 1000 Samples in 8 Hours." *IEEE/ASME Transactions on Mechatronics* 5, no. 2 (2000): 212-20.

(Meldrum, 2001a) Meldrum, Deirdre R., William H. Pence, Stephen E. Moody, David L. Cunningham, Mark Holl, and Peter J. Wiktor. "Automated Microfluidics for Genomics." *2001 Proceedings of the 23rd Annual EMBS International Conference* Istanbul, Turkey (2001) 2923-926.

(Meldrum, 2001b) Meldrum, Deirdre R., William H. Pence, Stephen E. Moody, David L. Cunningham, Mark Holl, Peter J. Wiktor, Mohan Saini, Matthew P. Moore, and Ling-Sheng Jang. "Automated, Integrated Modules for Fluid Handling, Thermal Cycling and Purification of DNA Samples for High-Throughput Sequencing and Analysis." *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings* Corno, Italy (2001): 1211-219.

Meldrum, D., C. Fisher, M. Moore, M. Saini, M. Holl, W. Pence, S. Moody, D. Cunningham, and P. Wiktor. "ACAPELLA-5K, A High-Throughput Automated Genome and Chemical Analysis System." *Proceedings of the 2003 IEEE Intl. Conference on Intelligent Robots and Systems* Las Vegas, NV (2003): 2321-328.

Merriam-Webster "Supernatant." Merriam-Webster. Incorporated, 2014. Accessed November 10, 2014. http://www.merriam-webster.com/dictionary/supernatant.

Metz, Trisha. "Lab Procedure." Integrated Circuit Fabrication. January 1, 2005. Accessed November 1, 2010. https://www3.nd.edu/~gsnider/labs446/Lab_procedure.pdf.

Najmabadi, Peyman, Andrew A. Goldenberg, and Andrew Emili. "Conceptual Design for an Automated High-Throughput Magnetic Protein Complex Purification Workcell." *Journal of Laboratory Automation* 8, no. 6 (2003): 101-06.

Najmabadi, Peyman, Andrew A. Goldenberg, and Andrew Emili. "A Scalable Robotic-based Laboratory Automation System for Medium-sized Biotechnology Laboratories." *International Conference on Automation Science and Engineering,* Proceedings of the 2005 IEEE, Edmond Canada, August 2005.

(Najmabadi, 2006a) Najmabadi, Peyman. *New Laboratory Automation System Architectures for Biotechnology Applications*. PhD thesis, University of Toronto, 2006.

(Najmabadi, 2006b) Najmabadi, Peyman, Andrew A. Goldenberg, and Andrew Emili. "A System Concept for Flexible Implementation of Robotic-based Automation in Biotechnology Research." *International Conference on Robotics and Automation,* Proceedings of the 2006 IEEE, Orlando, Florida, May 2006.

(Najmabadi, 2006c) Najmabadi, Peyman, Andrew A. Goldenberg, and Andrew Emili. "Hardware Flexibility of Laboratory Automation Systems: Analysis and New Flexible Automation Architectures." *Journal of Laboratory Automation* 11, no. 4 (2006): 203-06.

Ngatchou, P. N., Holl, M. R., Fisher, C. H., Saini, M. S., Dong, J., Ren, T. T. H., Pence, W. H., Cunningham, D. L., Moody, S. E., Donaldson, D. A., and Meldrum, D. R. “A Real-Time PCR Analyzer Compatible with High-Throughput Automated Processing of 2-µl Reactions in Glass Capillaries”. *IEEE Transactions on Automation Science and Engineering*, Vol. 3, No. 2, 2006, 141-151.

NuGEN. “System Uses Simple, Same-day Process to Give Researchers Global Gene Expression View Directly from Cell Lysate”. NuGEN, 2011, http://www.nugeninc.com/

QIAGEN. *High-throughput purification of PCR products using a centrifuge*. QIAGEN, 1997.

(QIAGEN, 2001a) QIAGEN. *Special Considerations for Isolation of RNA from Different Sample Sources*. QIAGEN, 2001.

(QIAGEN, 2001b) QIAGEN. *Stabilization of RNA in Biological Samples*. QIAGEN. 2001.

QIAGEN. *RNeasy® 96 BioRobot® 9604Handbook*. QIAGEN, 2002.

QIAGEN. *QIAquick® Multiwell PCR Purification*. QIAGEN, 2003.

(QIAGEN, 2006a) QIAGEN. *DNeasy® Plant Handbook*. QIAGEN, 2006.

(QIAGEN, 2006b) QIAGEN. *RNeasy® Mini Handbook*. QIAGEN, 2006.

(QIAGEN, 2007a) QIAGEN. *AllPrep® DNA/RNA/Protein Mini Handbook*. QIAGEN, 2007.

(QIAGEN, 2007b) QIAGEN. *QIAamp® DNA Mini and Blood Mini Handbook*. QIAGEN, 2007.

(QIAGEN, 2008a) QIAGEN. *AllPrep® 96 DNA/RNA Handbook*. QIAGEN, 2008.

(QIAGEN, 2008b) QIAGEN. *TissueLyser Handbook*. QIAGEN, 2008.

QIAGEN Sample & Assay Technologies. Accessed September 1, 2012. http://www.qiagen.com/.

Seethala, R. *Signal Detection Platforms for High-Throughput Screening*. Drug Discovery, Bristol-Myers Squibb, 2008.

Singh, Nanua. *Systems Approach to Computer-Integrated Design and Manufacturing*. 1st ed. New York, NY, John Wiley & Sons, 1996.

Soto, C., Saborio, G., Anderes, L. “Cyclic amplification of protein misfolding: application to prion-related disorders and beyond.” *Trends in Neurosciences*, Volume 25, Issue 8, 390-394, 1 August 2002.

Suh, Nam P. *Axiomatic Design Advances and Applications*. New York, NY, Oxford University Press, 2001.

Tecan. *HTS-TRACTM Compound Screening in Cells Flexible TRAC System for Multipurpose Cell Based Screening Assays*. Männedorf Switzerland: TECAN, 2008.

Thermo Scientific. "DNA Purification System." DNA Purification System. Accessed November 10, 2014. http://www.thermoscientific.com/content/tfs/en/product/dna-purification-system.html.

University of Washington "How Do Cells Connect with One Another?" The Naked Scientists. February 1, 2009. Accessed November 10, 2014. http://www.thenakedscientists.com/forum/index.php?topic=18902.0.

Wikipedia. "List of Purification Methods in Chemistry." Wikipedia. October 13, 2014. Accessed November 10, 2014. http://en.wikipedia.org/wiki/List_of_purification_methods_in_chemistry.

Zhang, Yaowu. *Group Technology: Manufacturing Cell Formation Incorporating Routing Flexibilities with Alternative Machines to Minimize the Material Handling Cost*. PhD Thesis, University of Houston, 2010.

Zymark Corporation. *Laboratory Robotic Handbook*. Hopkinton, MA, Zymark Corporation, 1988.