May 2020

A Reinforcement Learning Approach to Sequential Acceptance Sampling as a Critical Success Factor for Lean Six Sigma

Hani A. Khalil
University of Wisconsin-Milwaukee

A REINFORCEMENT LEARNING APPROACH TO SEQUENTIAL ACCEPTANCE SAMPLING AS A CRITICAL SUCCESS FACTOR FOR LEAN SIX SIGMA

by

Hani Abdulwahab Khalil

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin-Milwaukee

May 2020

ABSTRACT

A REINFORCEMENT LEARNING APPROACH TO SEQUENTIAL ACCEPTANCE SAMPLING AS A CRITICAL SUCCESS FACTOR FOR LEAN SIX SIGMA

by

Hani Abdulwahab Khalil

The University of Wisconsin-Milwaukee, 2020
Under the Supervision of Associate Professor Wilkistar A Otieno

In the 21st century, globalization coupled with technological advancement and free trade has created competition among various businesses enterprises. This competition has led many businesses to adopt various management techniques such as acceptance sampling aimed at transforming their processes in order to remain competitive in the global market and adapt to new market demands. The successful implementation of acceptance sampling is highly dependent on what the academic literature refers to as acceptance sampling optimization. A literature review on the optimization of acceptance sampling has not shown any work that studied whether acceptance sampling and machine learning (ML) plans can be considered as an optimal acceptance sampling technique (sequential sampling being one improved acceptance sampling technique). ML algorithms can be divided into four categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Reinforcement learning is different from the other types of machine learning, since it is a method of self-learning and acting based on observed data.

The aim of this dissertation is to develop a model based on coupling reinforcement learning methodology (RL) and sequential acceptance sampling in manufacturing to improve and achieve optimality in process and product monitoring. This model will serve as a continuous improvement strategy towards a better acceptance sampling implementation in the manufacturing industry. Simulation has been used as the model for proof of concept. The simulation model is designed to simulate any manufacturing process. However, this dissertation focuses on simulating the inspection process in a production line. In order to determine if an RL-based sequential sampling model is able to reduce the sample size and time of inspection, this dissertation compares the proposed model with the sequential acceptance sampling plan and the MIL-STD 1916

The result of the research will show the integration of sequential sampling and RL as a key to reduce the sample size and the sampling time interval during the inspection process in a manufacturing industry.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS, SYMBOLS, and VARIABLES

α	maximum allowable Type I estimation error (producer's risk)
β	maximum allowable Type II estimation error (consumer's risk)
γ	parameter of a Beta distribution representing pseudo-failures
γ	parameter of an initial prior Beta distribution representing pseudo-failures
λ	discount factor
N	lot or batch size
n	sample size
c	acceptance number
d_1	nonconforming items / defectives items
r_1	rejection number of the first sample
r_2	rejection number of the second sample
c_2	cumulative acceptance number of both first and second samples
Ho	null hypothesis
H_a	alternative hypothesis
p	quality parameter of a lot under inspection
p_1	acceptable quality level of nonconformities
p_2	rejection quality level of nonconforming units
X_a,	acceptance line
X_r	rejection line
s	slope
k	average acceptable number of defective unites.
R	agent's reward
S	agent's state
A	agent's action
H	agent's history
w	wait time
π	agent's policy
V	state value function
Q	action-value function
t	variable, counter, number of replications
G_t	total future rewards
k	variable, counter, number of replications
LSS	lean six sigma
KPO	key process output
KPI	key process indicators
AI	artificial intelligence
ML	machine learning
RL	reinforcement learning

ASQ	American Society for Quality
AQL	acceptable quality level
RQL	rejectable quality level
SARSA	state-action-reward-state-action
SS	six sigma
MAIC	measure, analyze, improve and control
DMAIC	define, measure, analyze, improve and control
CI	continuous improvement
DOD	The Department of Defense of the United States
MIL-STD-105E	military standard-105E
MIL-STD 1916	military standard-1916
SPRT	sequential probability ratio test
MDP	Markov decision process
POMDPs	partially observable Markov decision processes
FHT	fuzzy hypothesis testing

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my advisor, Dr. Wilkistar Otieno, for her continuous support of my dissertation, her patience, her motivation, and her immense knowledge. Her guidance has helped me in each phase of research and writing this dissertation. I could not have imagined having a better advisor for my dissertation study. Besides my advisor, I would like to thank the rest of my dissertation committee members, Dr. Hamid Seifoddini, Dr. Matthew Petering, Dr. Rohit J. Kate, and Dr. Xiaohang Yue for their insightful comments and encouragements.

I would like also to thank my friend Eric Amoah Sakyi for his assistance in the research. I gratefully indebted to him for his very valuable comments and assistance on this work.

Finally, I must express my very profound gratitude to my parents, my wife, my brothers and sisters for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation. This accomplishment would not have been possible without them.

Thank you all for your support.

Chapter 1: Introduction

The goal of Chapter One is to enable the reader to have an overview of what this thesis all about by covering the research framework, problem statement of the study, objectives of the research, the research questions, and the significance of the study.

1.1 Research Framework

The goal of this dissertation is to integrate the Reinforcement Learning RL tool as a solution framework to Sequential Sampling in a Lean Six Sigma methodology (LSS) of process management. The result of the dissertation will show the integration of sequential sampling and reinforcement learning as an optimal acceptance sampling technique in a manufacturing industry. This paper is divided into chapters; the soul of this research will be defined in Chapter One by way of problem statements, the objective of the research, the research questions, and the scope of study.

Chapter Two, on the other hand, will explain and provide the necessary background information on which the research was assessed for suitability. The fundamentals of the industrial and manufacturing revolutions and the history of modern quality and quality control methods as part of the continuous improvement strategy have also been discussed in Chapter Two. Lean six sigma (LSS) methodology is also introduced, with emphasis given to its strengths and weaknesses. Acceptance attribute sampling plans are also discussed as a part of LSS tools, focusing on a sequential acceptance sampling plan. The end of Chapter Two includes a discussion of the background of Reinforcement Learning as one of the current topics in Machine Learning (ML). The literature review section in Chapter Three will include a discussion of past research that address sequential sampling methodologies, and reinforcement learning with some applications. Most importantly, the literature review will show a dearth of past work on RL application to sequential acceptance sampling, which is the main contribution of this dissertation. Chapter Four will give a summary of the analyzed literature review and the linkage between reinforcement learning and sequential acceptance sampling. Chapter Five will present the methodology of sequential acceptance sampling based on RL. Chapter Six will simulate the presented methodology using an open-source software called Python and comparing its results with the results of simulated acceptance sampling plans.

1.2 Problem Statement

For the past decades, globalization coupled with technological advancement and free-trade has created competition among various businesses enterprises. This competition has led many businesses to adopt various quality management techniques such as acceptance sampling, aimed at transforming their processes in order to remain competitive in the global market and adapt to new market demands (Alhuraish, Robledo, & Kobi, 2017).

Although many studies show the success stories of acceptance sampling, they also portray some of the barriers that tend to deter companies from deploying acceptance sampling. Some of the barriers that prevent companies from enjoying the merits of sampling include the risk associated with rejecting "good" lots or accepting "poor" lots (Mitra, 2016).

The problem is that even though a lot of academic research has focused on improving acceptance sampling plans, particularly sequential sampling, none has delved into the importance and effectiveness of coupling reinforcement learning with sequential acceptance sampling to serve as a new model to reduce sample size and eventually the production cost

1.3 Goals of Research

The overarching goal of this research is to examine and develop a model based on coupling reinforcement learning methodology (RL) and sequential acceptance sampling in manufacturing to improve and achieve the optimal sample size and product monitoring. This model will serve as a continuous improvement strategy towards a better acceptance sampling implementation in the manufacturing industry. In other words, this dissertation aims to develop an improved acceptance sampling plan that incorporates reinforcement learning in the quality decision-making framework as shown in Figure 1.

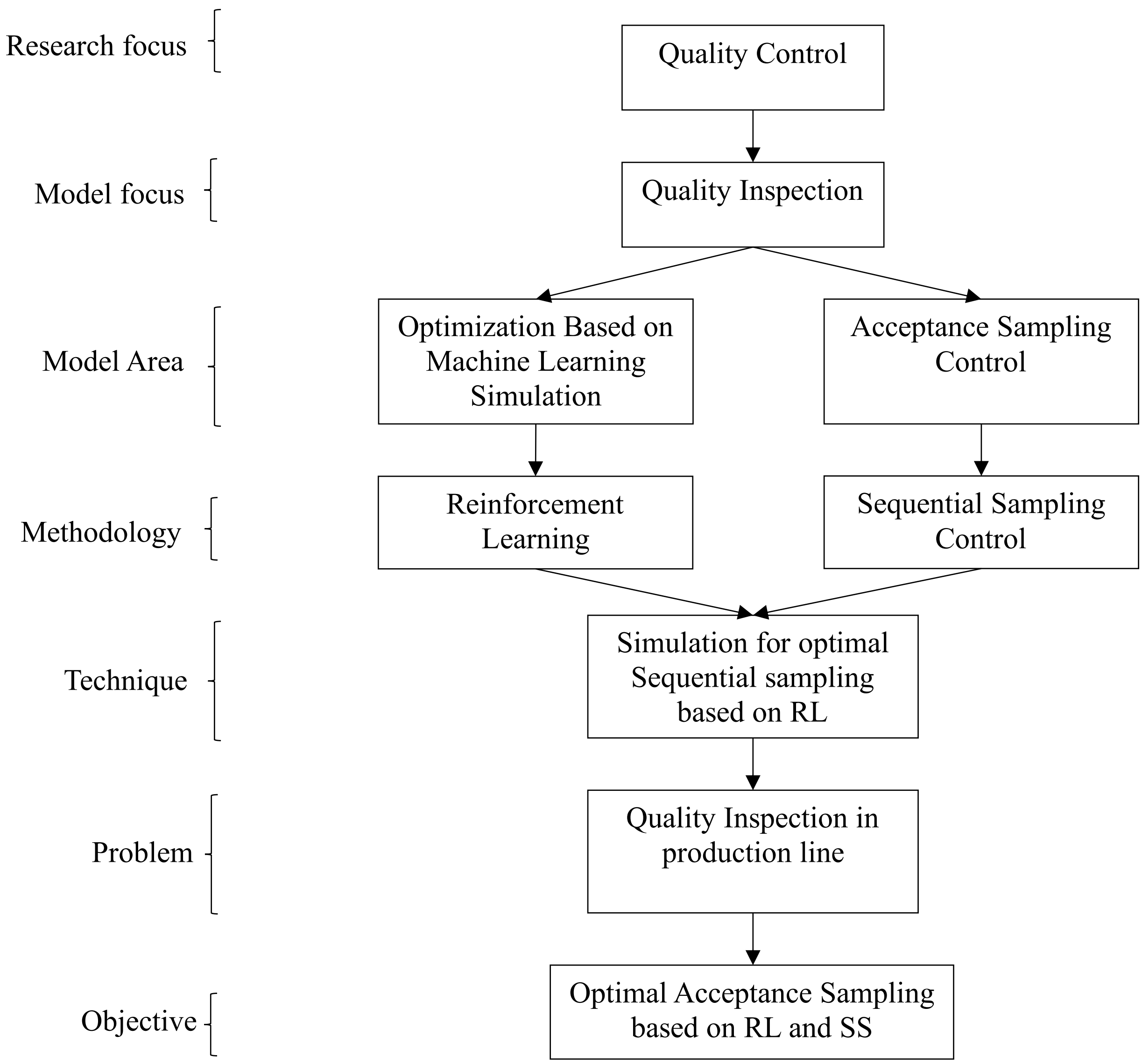


Figure 1: General framework for the proposed model

1.4 Research Questions

The primary research questions are the following:

I. Can reinforcement learning be applicable to acceptance sampling?

II. Is there a difference between the traditional sequential sampling and sequential sampling using the reinforcement learning methodology?

The secondary research questions are the following:

I. Is RL part of ML techniques in the fourth industrial revolution?

II. What are the latest techniques for optimizing RL and sequential acceptance sampling?

1.5 Significance of Study

Even though the current massive studies on sequential acceptance sampling plans have provided significant insight for successful acceptance sampling implementation, there is still an opportunity to identify more techniques for optimizing the implementation of acceptance sampling. The significance of this research is that the outcome of the study may prove important in contributing towards the success of acceptance sampling implementation in manufacturing by identifying and adding a new technique to the current acceptance sampling plans.

The excessive demand for implementing acceptance sampling plans in manufacturing justifies the need for more research into acceptance sampling. From the sequential sampling standpoint, the results of this research will ultimately reduce the number of samples needed to be taken for quality determination and control, hence addressing the disadvantage of the high cost of quality implementation in the process of inspection.

The following chapter provides a background into the history of manufacturing and quality control, which will lead to the need for the proposed research methodology as a road map for further studies.

Chapter 2: Research Background

2.1 Introduction

Chapter Two contains three parts, as follows: a background of how the manufacturing revolution played a major role in the industrial revolution and how quality has been integral in the manufacturing revolution. A background of acceptance sampling plans is also presented. Lastly, a background of how Reinforcement Learning became a major topic in manufacturing is followed by a brief chapter summary.

2.2 Background part 1: Brief discussion of the Impact of Manufacturing in the American Economy

In today's global economy, manufacturing continues to be a cornerstone for creating jobs for millions of people living in developed and developing nations. The manufacturing sector remains the major driving force for economic growth. Manufacturing activities have not only created lasting wealth, but they have also more importantly aided in distributing wealth in both developed and developing nations by creating high-paying jobs. Higher wages translate into changes in the economic status of the working class in society to either middle class or upper middle class (United Nations Industrial Development Organization, 2015).

It is important to point out that manufacturing activities in recent decades have seen major shift in the regions where certain products are made. For example , the majority of companies in the United States are moving their low-tech manufacturing activities overseas where cost of labor and raw material is relatively cheaper. The transfer of such internal company activities to another country is referred to as offshoring. On the other hand, moving some of the internal manufacturing activities outside the company is referred to as outsourcing. If the manufacturing activities are moved back to the U.S., it is referred to as reshoring and if manufacturing activities are moved nearer the U.S., then it is termed as nearshoring. The driving force behind these relocations of manufacturing activities includes, but is not limited to, cost reduction, quality improvement, and productivity improvement (Hartman, Ogden, Wirthlin, & Hazen, 2017). It is important to note that developed nations have moved to manufacturing very sophisticated technologies in the aviation, medical and cyber-security arenas. That is to say that employment growth in the manufacturing sector has not been uniformly distributed among countries.

The manufacturing industry has evolved through the industrial revolution, which in turn led to the evolution of several manufacturing paradigms since its beginning over two centuries ago (World Economic Forum, 2016). The next section will provide some background of the different stages of the industrial revolution, focusing on their influence in the manufacturing sector.

2.2.1 Industrial Revolution and The Manufacturing Revolution

The industrial revolution that spanned from the late 1760s to date triggered the manufacturing revolution. The first industrial revolution (1780-1870) was ignited by the construction of railroads and introduction of steam engines that powered the majority of production during this era (Hudson, 2014). The first industrial revolution not only made mechanical production possible, but it also more importantly created a major shift in employment and income from the agricultural to industrial activities (Trew, 2014). The Second Industrial Revolution (1870-1970), on the other hand, was ushered in by the introduction of electricity and the assembly line. The introduction of this new science (electricity) and process improvement methods (assembly lines) helped to streamline the production of commodities, thereby making mass production feasible in the late 19^{th} and early 20^{th} centuries.

The digital or computer phase, also called the third industrial revolution, began in the 1970s and was initiated by the introduction of semiconductors, mainframe computing, personal computing and the internet (Schwab, 2016). The third industrial revolution took place as a result of extensive use of electronics and information technology, thereby allowing the introduction of automation in the manufacturing industry. This revolution offered a solution to a mass production paradigm, which was running into difficulties in relation to the need for increased production as a result of increased population growth rate, high levels of unemployment, and global trade imbalances. Automation allowed a transition from an era of mass production to one of flexible specialization that allowed for some degree of customized manufacturing.

The current fourth industrial revolution, which began in 2006, is characterized by the use of Cyber-Physical Systems (CPSs) resulting from the fusion of technologies that are blurring the lines between the physical, the digital, the cyber and the biological spheres (Régio, Gaspar, Farinha, & Morgado, 2016). The term CPSs emerged in the United States in 2006 and is attributed to Dr. Helen Gill who was then at the National Science Foundation. A cyber-physical environment can be defined as "the interaction of computation with a physical process, usually with a feedback loop where physical process affects computation and vice versa" (Herwan, Kano, Oleg, Sawada, & Kasashima, 2018). The fourth industrial revolution helped trigger a wave of breakthroughs in areas ranging from gene sequencing to nanotechnology and from renewables to quantum computing. It is the fusion of technologies and their interaction with the physical, digital and biological domains that make the fourth industrial revolution fundamentally different from previous revolutions. The establishment of the fourth industrial revolution paved the way for more advancement in technology such as automated vehicles, 3D Printing, advanced robotics and new and self-healing materials, just to name a few. In particular, advanced robots would not exist without the added values of the current era of artificial intelligence and machine learning (ML), which largely depends on high power computing capabilities (Syam & Sharma, 2018; Wood et al., 2017). The industrial revolution contributed significantly in creating what has become in today's global market. Figure 2 depicts the timeline of the four industrial revolutions.

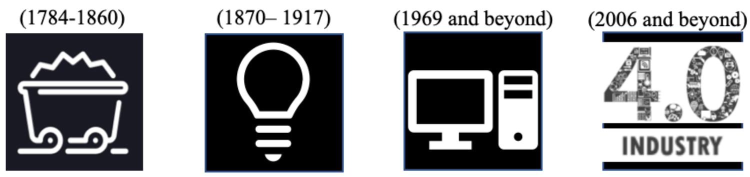


Figure 2: Industrial revolutions' path

The next section will discuss the major paradigms in consumer goods manufacturing, namely (1) craft production, (2) mass production, (3) mass customization and (4) global manufacturing (Hu, 2013).

2.2.1.1 Craft Production Paradigm

Craft production dates back to the first industrial revolution and is attributed to creating exactly the product the customer asks for, on demand and usually one unique product at a time using specializing hand tools in small machine shops (D. Chen et al., 2015). There were no manufacturing systems associated with craft production and this form of production was not scalable since the majority of the craft laborers, who had unique knowledge, were confined to specific geographical areas. Craftsmanship, which generally used highly skilled labor, was used in making precision parts used in automobile and drive trains. This manufacturing paradigm was not capable of handling the demand for high volumes of products. With the emergence of new markets and new technologies, the craft production paradigm evolved into mass production through the interchangeability and the moving assembly lines (D. Chen et al., 2015; Hu, 2013).

2.2.1.2 Mass Production Paradigm

This paradigm was widely used in the 20th century during the second industrial revolution. It is defined as the production of extremely large quantities of identical products for a specific period of time. This manufacturing paradigm was carried out through synchronized flow of production lines to manufacture key precision hardware and assembly of the finished product. To enable response to the high demands of products from consumers, companies had to maintain high production volumes and as a result had to incorporate machinery into their production system to take the place of human labor. The main objective of this paradigm was to reduce the cost of manufacturing, which translated into low product price (Koren, 2010).

2.2.1.3 Automated Production and Mass Customization

Mass customization may be defined as the production of an expanded variety of products of the same product family at a low cost. This manufacturing paradigm began in the 1980s and was initiated by society's need for a larger product selection. The key characteristic is that manufacturers tend to offer product "options" that add extra features to the standard product. The main objective of this manufacturing paradigm is to increase the variety of products at a low cost to the customer (K. Chen, 2014). The manufacturers have full control of the basic product options that they can offer, and customers choose the "package" that they want, buy it and only then is the product moved to the next stage to be finished according to custom choices. This approach allows a manufacturer to use the unique strengths of its mass production resources to produce major components of a particular product at the lowest cost while leaving the customization process to be the last step in the manufacturing process of the products with optioned accessories. Mass customization has been made possible by the use of computers in industrial manufacturing, which has in turn made it possible for flexible automation that helps make mass customization inexpensive (Koren, 2010). One may argue that mass customization has a similar concept like craft production. However, is important to make it clear that mass customization does not necessarily mean producing a unique product as expressed in the craftsmanship form of production.

2.2.1.4 Personalized Customization and Digital Technology

The emergence of the internet, smart computers and technologically advanced manufacturing systems such as 3D printing has made it possible for a new manufacturing paradigm to be born, known as personalized customization (Sridharan, 2015). The basis of personalization is to tailor contents to known wishes and needs of a customer. This manufacturing paradigm requires the participation of consumers in the design, product simulation, manufacturing, supply and assembly process of goods and services to meet consumer needs and preferences (Sridharan, 2015). Under this manufacturing paradigm, customers create various innovative products and realize value by collaborating with manufacturers and other consumers and this process is enabled mainly by open product architecture. A summary of the four manufacturing paradigms is depicted in Figure 3.

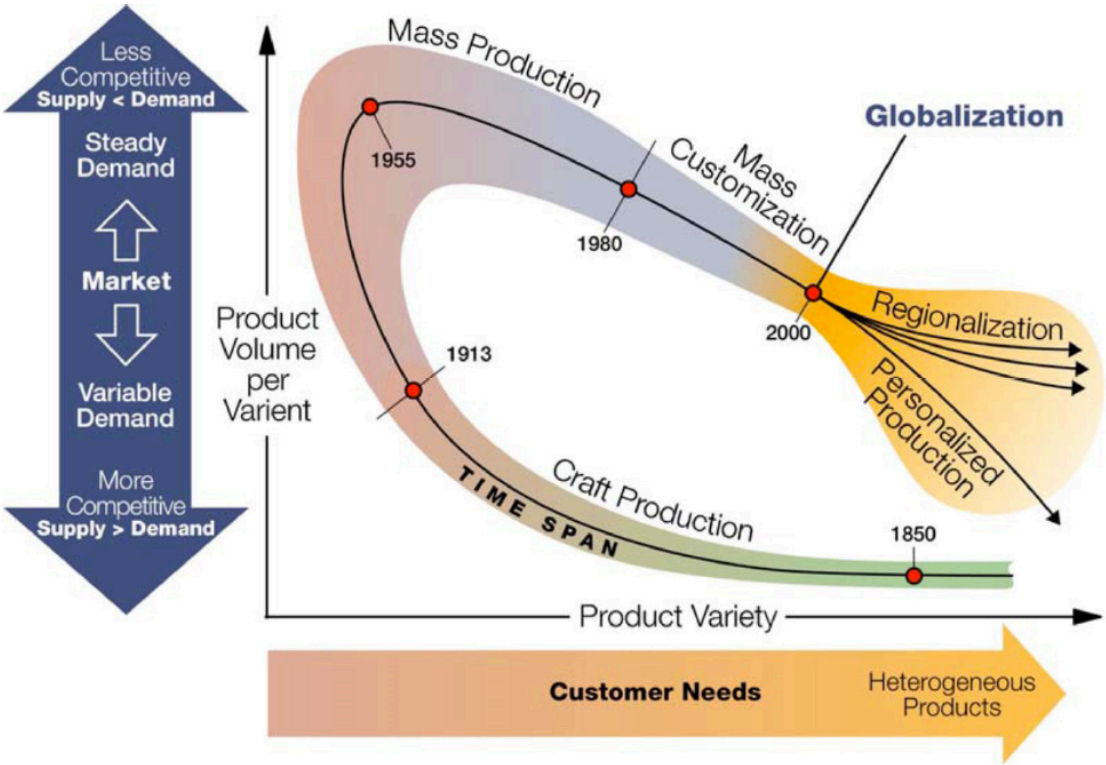


Figure 3: The drivers to new paradigms (Koren, 2010)

2.2.2 Quality and its Importance to Manufacturing

The term quality may simply refer to the degree of excellence of a product or service to a customer or the end user of a produced part or product. Quality has a long history and its application has accelerated over the last few decades. Joseph M. Juran, who is an honorary member of ASQ, concluded that the application of quality dates back to the ancient Egyptians and the building of the pyramids (Borror, 2009). Prior to the industrial revolution, quality was heavily associated with craftsmanship where each craftsman had full control of the end product of his craft (Borror, 2009). This meaning of quality changed with the emergence of the industrial and manufacturing revolution. For the past decades, the concept of quality control has evolved to become a key strategic tool in operating a successful business enterprise in a competitive global market.

The motivation and rationale behind quality in every business enterprise stems from the increasing demand from consumers for companies to produce goods that meet or exceed their needs and expectations. For countries such as the United States, China, Japan, Germany, and the United Kingdom, just to mention but a few, to continue their strong economic power, their manufacturing sectors must remain competitive (Gray, 2017). One may argue that economic indicators such as the national trade deficit are often linked to a country's manufacturing capabilities and competitiveness. Based on multiple studies, customers are willing to pay more for a better quality of product, hence making quality in manufacturing a critical factor for nations such as the United States to reduce their trade imbalances with other nations.

Striving for quality has always been and will always be a major part of human endeavor. The implementation of quality takes two major approaches. The first approach seeks to take the attitude that quality is fundamental to all functions, policies, and procedures of a company. The second approach, on the other hand, seeks to implement quality on an "as–needed" basis, which implies adding quality to a product or service as a means to improve and maximize profits through a highly specialized team (Jones, 2014). In most cases, companies use both approaches throughout their business operations in an effort to produce products that meet customer expectations.

2.2.3 Cost of Quality

The majority of business enterprises may attest to the fact that good quality products and services go hand-in-hand with cost associated with satisfactory products and services. For the past decades, one of the key stumbling blocks to the creation of stronger quality programs was linked to the misguided idea that the achievement of better-quality products required much higher costs. Poor quality means poor resource use that may involve waste of materials, poor use of labor and equipment time, which then leads to higher costs (Feigenbaum, 1991).

On the other hand, one may attribute satisfactory quality to satisfactory resource use that then leads to lower costs. One of the major factors in these misguided past notions of the relationship between quality and cost was the unavailability of meaningful data since businesses in the past believed that quality could not practically be measured in terms of cost. The reasoning behind this belief is the fact that traditional cost accounting failed or did not attempt to quantify quality, which may have to do with the fact that quality cost did not easily fit into older accounting structures for most businesses (Feigenbaum, 1991).

In today's global market, businesses not only recognize the measurability of quality costs but that these costs are vital to the management and engineering of modern total quality control as well as to the business strategy planning of companies and plants. The costs of quality can be defined as the basis through which investments in quality programs may be evaluated in terms of cost improvement, profit enhancement, and other benefits for plants and companies from these programs (Feigenbaum, 1991). It is also important to point out that the cost of quality is generated not only throughout the marketing, design, manufacturing, inspection shipping stages or cycle but also continues to be accounted for throughout the total life cycle of the product in service and use.

Accounting for the cost of quality in a business enterprise includes two major areas: the costs of control and the costs of failure of control. These are known as producer *operating quality costs*. From the book, (Total quality control), *operating quality costs* entail those costs associated with the definition, creation, and control of quality as well as the evaluation and feedback of conformance with quality, reliability, and safety requirements, and those costs associated with the consequences of failure to meet the requirements both within the factory and in the hands of customers (Feigenbaum, 1991).

According to Feigenbaum, the costs of control can be accounted for in two main segments, namely *prevention costs and appraisal costs. Prevention costs* help keep defects and nonconforming products from occurring and include the quality expenditures to keep unsatisfactory products from ending up in the hands of customers. Examples of these costs may include employee quality training and quality engineering. *Appraisal costs,* on the other hand, involve the costs for maintaining company quality levels by means of formal evaluations of product quality. Examples of such costs are inspections, tests, outside endorsements and quality audits. The costs of the failure of control that are attributed to products not meeting the quality requirements and poor materials can also be placed in two segments, namely *internal failure costs* and *external failure costs. Internal failure costs* include scraps, spoilage and reworked material. *External failure costs* include product performance failures and customer complaints (Feigenbaum, 1991).

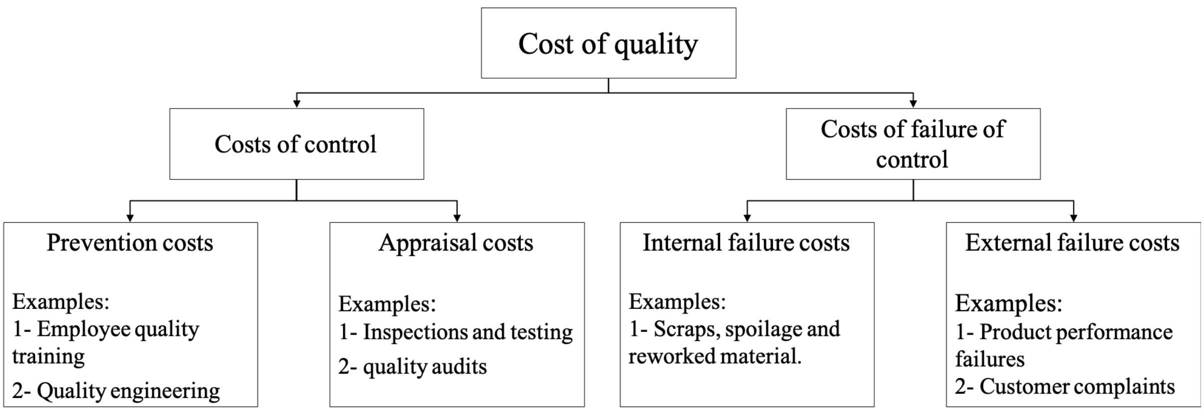


Figure 4: Types of the costs of quality

2.2.4 The History of Modern Quality

The implementation of modern process management principles has existed for centuries and its usage continue to evolve over time (Figure 5). These process management principles began with the scientific management, Toyota production (lean manufacturing), total quality management, ISO 9000, six sigma, and lastly, lean six sigma. These principles will be discussed briefly in this part of the dissertation.

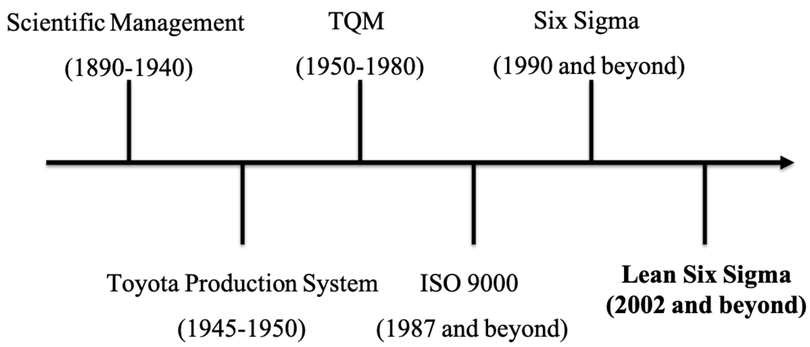


Figure 5: Timeline of modern quality strategies

2.2.4.1 Scientific Management (1890-1940)

Scientific management was introduced by Fredrick Taylor and applied in the manufacturing industry between 1890 and 1940. Scientific management refers to a management technique for improving work efficiency, labor productivity and standardizing work processes. It is considered to be a backbone of modern-day lean manufacturing. The main objective of scientific management is to enforce thorough collection of data, continual improvement to identify an optimal approach to conduct every manufacturing operation. It also exemplifies the need for management of an organization to monitor and enforce standard operating practices (Franchetti, 2015). Figure 6 is a representation of the scientific management as proposed by Taylor.

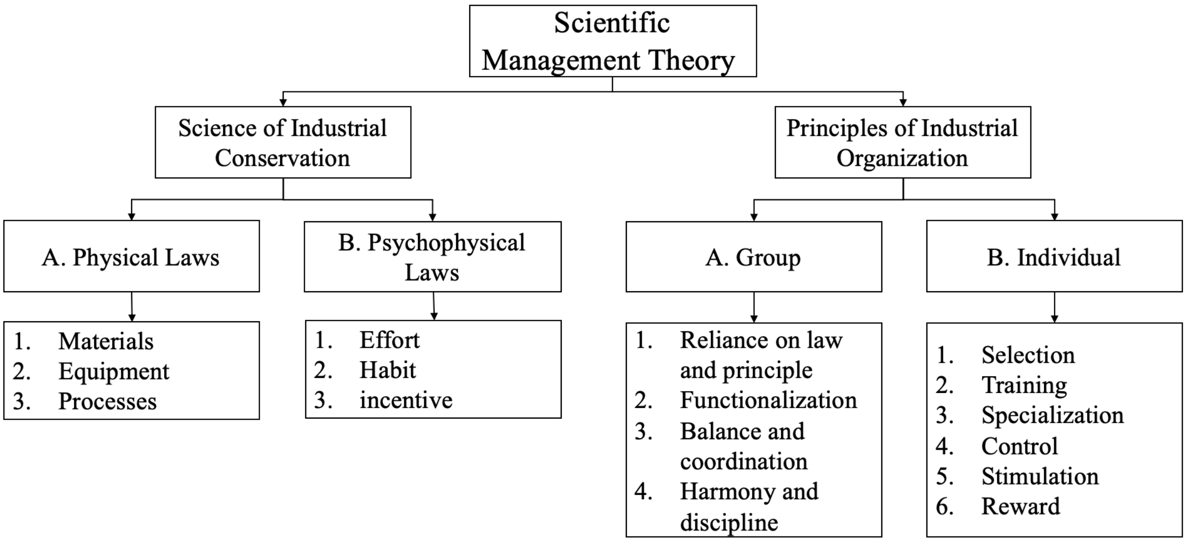


Figure 6: The Content of Scientific Management (Thompson, 1917)

2.2.4.2 Toyota Production System (1945-1950)

The Toyota Production System (TPS), on the other hand, was developed after World War II (1945-1950) by Taiichi Ohno and his associates who were working at Toyota Motor Corporation. Faced with a scarcity of resources, Eiji Toyoda, who later became the president of the Toyota Corporation, is largely attributed to have brought the company international eminence. Eiji Toyoda asked Taiichi Ohno (shop floor supervisor) and his associates to develop a mechanism to reduce waste in the production system. The quest led to the development of the TPS paradigm that was hinged upon the reduction of seven production wastes, namely, (1) over-production, (2) defects, (3) unnecessary inventory, (4) inappropriate processing, (5) excessive transportation, (6) waiting, and (7) unnecessary motion. These seven forms of wastes became the principal cornerstones in the development of the Lean and Just In Time (JIT) strategies (Pepper & Spedding, 2010). TPS was modeled to shine light on the end customer and create processes that drive end value and consistent delivery for the customer. It uses the application of more efficient production systems to eliminate material waste and time (Franchetti, 2015). That is to say, TPS sheds more light on the end customer because it aims to (1) provide outstanding quality products and service to the customer, (2) develop a culture within the company where mutual respect, trust and cooperation are embraced, (3) eliminate waste, which in turn reduces cost and maximizes profit, and (4) create flexibility in production standards to meet market demand (Smalley, 2013). Figure 7 is a schematic representation of the TPS strategy and more insight can be obtained from (Liker, 2004; Stecher and Kirby, 2004).

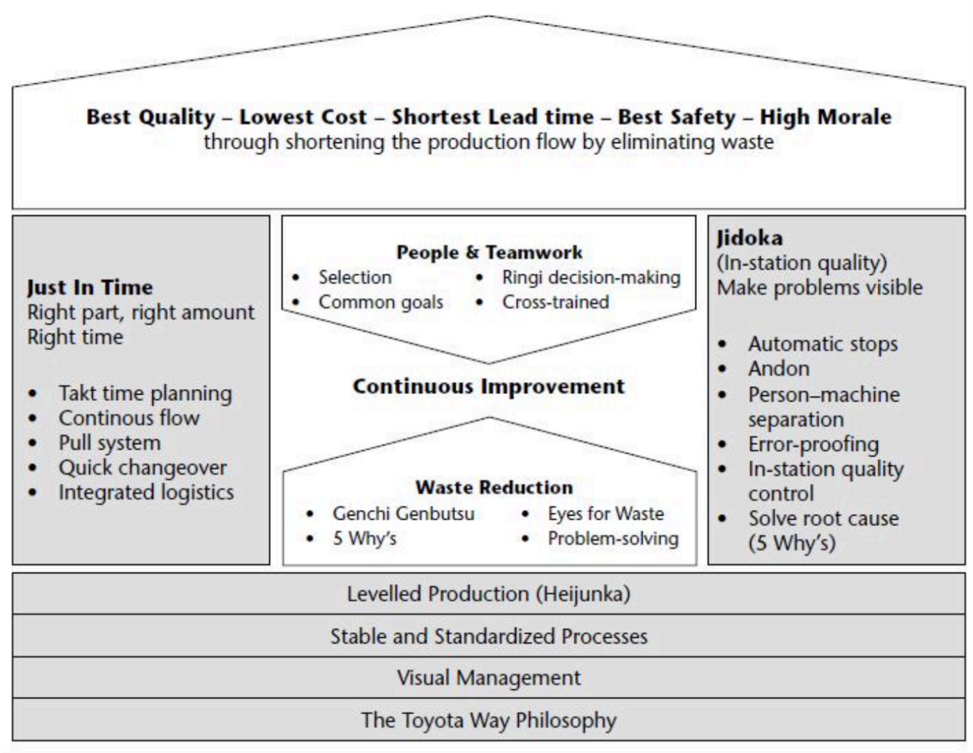


Figure 7: The Toyota Production System (Liker, 2004)

2.2.4.3 Total Quality Management (1950-1980)

Between 1950-1985, a new phase of a quality management principle emerged, known as Total Quality Management (TQM). The concept of TQM resurfaced in Japan after World War II when two American statisticians (Edward Deming and Joseph Duran) embarked on a journey to implement statistical quality control concepts in an effort to rebuild Japan's manufacturing companies (Franchetti, 2015). In 1951, Feigenbaum published the first edition of the book Total Quality Control where he established the principles of Total Quality

Management (TQM). TQM, according to Feigenbaum, “is an effective system for integrating the quality development, quality maintenance, and quality improvement efforts of the various groups in an organization so as to enable production and service at the most economical levels which allow full customer satisfaction” (Feigenbaum, 2019). Figure 8 is a representation of the TQM strategy and more insight can be found in (Borror, 2009; Feigenbaum, 1991; Naidu, Babu, & Rajendra, 2006; Zairi, 1991).

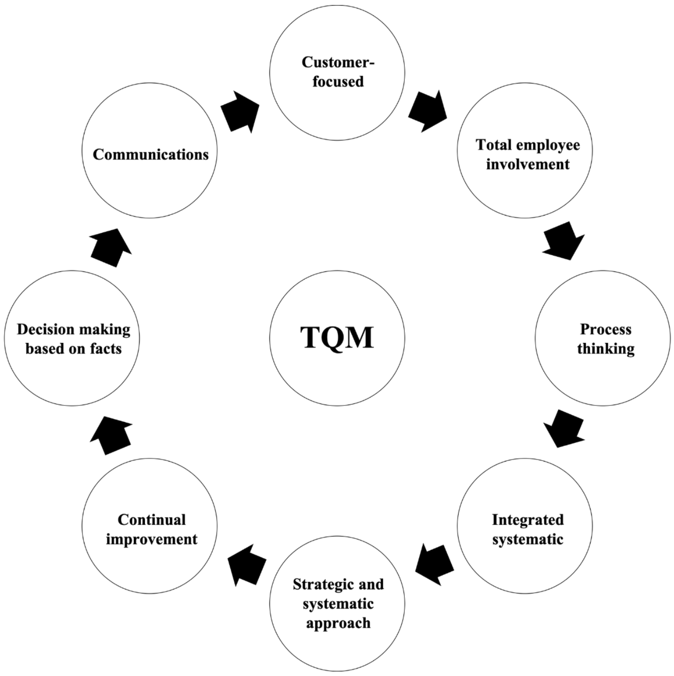


Figure 8: Total Quality Management Elements (ASQ, 2019)

2.2.4.4 ISO 9000 (1987 and Beyond)

ISO (International Organization for Standardization) is known to be the world’s largest developer of voluntary international standards used around the world. It was founded in 1947 right after World War II. These standards are written with the mindset that a well-designed quality assurance program provides confidence in a company’s product and management.

In 1987 the international organization with its 91-member nations adopted the ISO 9000 series of standards and its implementation in manufacturing industries began. However, in

1994 the standard was updated to an ANSI/ISO/ASQ standard to enable a majority of U.S. companies to be certified to the standard. Many companies around the world continue to adopt the ISO 9001 Standard mainly because in the 21st Century, companies that are not ISO 9001 certified find it difficult to obtain contracts from other companies or from their respective governments. The main driving force behind adopting an international standard was to avoid conflict of different national standards that govern the production of goods and services ("The ISO story," 2017).

2.2.4.5 Six Sigma (1990 and Beyond)

The concept of Six Sigma (SS) is another of the management strategies many companies have adopted. It was developed by Motorola in 1987 and was later used and improved by General Electric (Taylor, 2014). The primary objective of this concept is to reduce product defects to assure the sustainability of an organization in a competitive global market. The term SS in manufacturing can be interpreted as 3.4 defects per million opportunities (DPMO), which means 99.99966 percent of the goods or services must be free of defects, and its focus is always on financially measurable results. Considering that many organizations have an error rate of between 35,000 and 50,000 in a million, their sigma level of error is within 3 and 4 standard deviations from the mean (Griffin et al., 2016). Hence, the range for improving a process to achieve the 6-sigma level is still enormous.

Table 1: The six sigma measurement (Ellis, 2016)

Sigma Level	Defects per Million Opportunities (DPMO)
1	691,462
2	308,537
3	66,807
4	6,210
5	233
6	3.4

Table 2: Another look of six sigma measurement

Sigma Level	% Non-Defective
1	30.9%
2	69.15%
3	93.32%
4	99.379%
5	99.9767%
6	99.99966%

The six sigma strategy is characterized by a sequence of predefined phases that include: Measure, Analyze, Improve and Control, and formed what was known as the MAIC process. These four phases were first used by Motorola around 1987. As the manufacturing sector progressed, around 1995, GE incorporated a fifth phase called Define, hence establishing the DMAIC process (Define, Measure, Analyze, Improve and Control) (Lal, Kumar, & Bhardwaj, 2014).

Table 3: The road map for DMAIC

Define	Measure	Analyze	Improve	Control
- Define the project	- Understand the process	- Evaluate risks on process inputs	- Verify critical inputs using planned experiments	- Finalize the control system
- Define the process	- Develop and evaluate measurement system	- Analyze data to prioritize key input variables	- Design improvements	- Verify long term capability
- Determine customer requirements	- Measure the current process performance	- Identify waste	- Pilot new process	
- Define key process output variables				

The DMAIC process is often used in the production sector of a business unit and considered to be a "Closed Loop" methodology that aided in the elimination of ineffective steps in production. Its primary focus is on measurement and improvement processes and it often uses technology for Continuous Improvement (CI). Continuous improvement may entail seeking to improve and enhance every process by being aware of the variations in each process in order to diminish those variations. Figure 9 is a flow Chart that explains the loop for the five phases and Table 3 elucidates what needs to be executed at each phase (Barone & Eva, 2012).

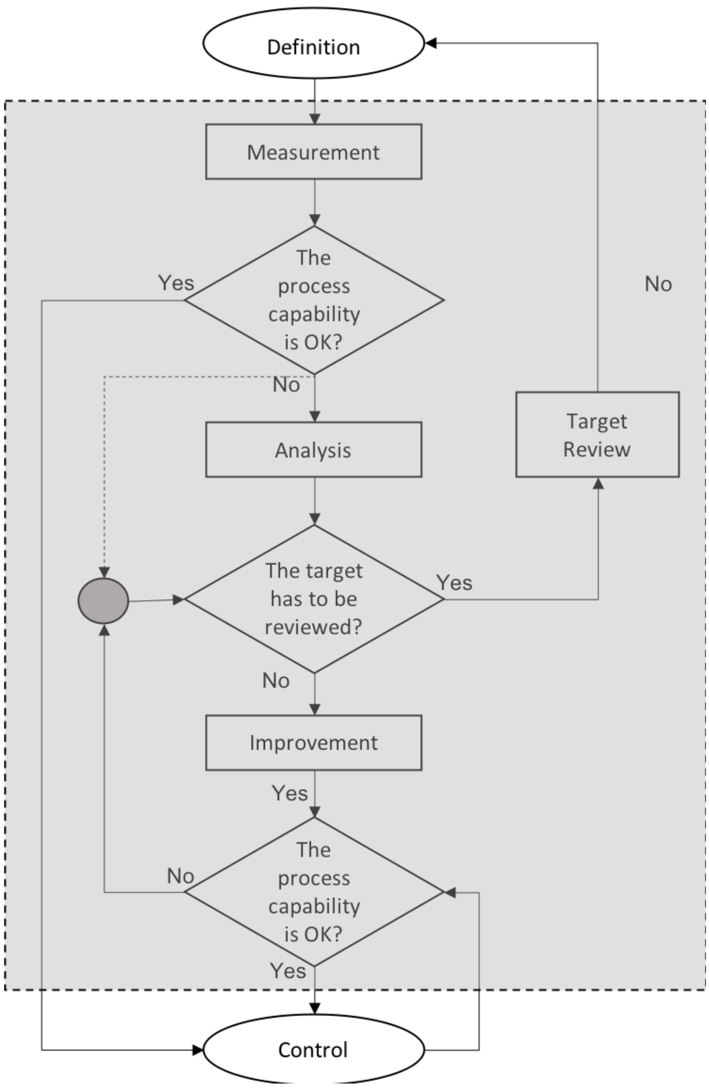


Figure 9: DMAIC Flow Chart (Barone & Eva, 2012)

2.2.4.6 Lean Six Sigma (2002 and Beyond)

The concept of lean six sigma is the latest modern quality principle, introduced in 2002 by Michael George in the book *Lean Six Sigma: Combining Six Sigma with Lean Speed.*
Lean six sigma, as the name implies, is a combination of lean and six sigma methodologies and focuses primary on quality improvements and cost reduction for businesses, processes or a product. According to Stern (2016), lean six sigma can also be defined as a "hybrid methodology" designed to accommodate global challenges and international constraints by capitalizing on two main powerful process improvement methodologies: six sigma and lean thinking (Stern, 2016).

Prior to the introduction of lean six sigma, leaders in different organizations viewed quality improvements and work efficient initiatives as two separate undertakings (Jones, 2014). Whereas the six sigma strategy focuses only on quality, the lean strategy addresses process speed and workflow efficiency. The introduction of lean six sigma therefore helped bridge the gap and brought some awareness in implementing both quality improvements and cost reduction initiative simultaneously to overcome the shortcomings of both methodologies when implemented separately (George, 2002). In recent years, LSS has been used as a continuous improvement tool in the manufacturing and service sectors to achieve quality and operational excellence in the business arena (Lal et al., 2014). These benefits include, but are not limited to (Albliwi, Antony, & Lim, 2015):

1. Increased profits and financial savings
2. Increased customer satisfaction
3. Reduced cost
4. Reduced cycle time
5. Improved key performance metrics

6. Reduced defects
7. Reduced machine breakdown time
8. Reduced inventory
9. Improved quality
10. Increased production capacity

Section 2.2.5 is entirely dedicated acceptance sampling plans as strategies for implementing LSS, the latest modern quality principle in manufacturing.

2.2.5 Data Acquisition and Acceptance in Lean Six Sigma

Data acquisition is a key characteristic in the successful implementation of LSS. The DMAIC cycle, as the name implies, uses statistical analysis and tools to define, measure, analyze, improve and control business processes. As it was stated earlier, the objective of LSS is to enhance the quality of a product or service (key process output) by minimizing variability of the key process indicators (KPI). Therefore, sufficient knowledge and control about process inputs (X's) will increase the accuracy in the prediction and control of process outputs (Y's). This is generally expressed as $Y = f(X_1, X_2, X_3, \dots, X_k)$.

To reduce variability in a process or production, it is critical to employ statistical methods to aid in the verification of the quality of the products or services at different stages of the process. For a process to be verified, businesses need to plan, collect, and analyze data on the sample that has been taken from the entire production. Without data acquisition it can be challenging to evaluate variability in the process. Some of the statistical methods used in LSS to achieve DMAIC objectives include: means and variance, random variables and probability distributions, control charts, sampling and acceptance sampling, hypothesis testing, design of experiments, regression analysis, reliability engineering, and tolerancing (Franchetti, 2015).

Table 4 illustrates the statistical tools in DMAIC and shows where acceptance sampling plans (sequential sampling), which are the focus of this research, are used in the LSS strategy.

Table 4: Statistical Tools for Each Phase in LSS

Define	**M**easure	**A**nalyze	**I**mprove	**C**ontrol
Project charter	Operational definitions	Pareto charts	Design of experiments	Control planning
PIP management process	Data collection plans	C & E matrix	Kanban/Pull	Process documentation
Value stream map	Pareto chart	Fishbone diagrams	TPM	Standard operating procedures
Financial analysis	Histogram	Brainstorming		Training plans
Multi generation plan	Box plot	Basic statistical tools	Process flow improvement	Communication plan
Stakeholder analysis	Statistical sampling	Constraint identification		Mistake proofing
Communication plan	Continuous MSA (Gage R&R)	Non-parametric	Replenishment pull	Statistical process control
SIPOC map	Attribute MSA (Kappa studies)	Confidence intervals	Sales and operation planning	Implementation plan
High-level process map	Control charts	Simple and multiple regression	Quick changeover	Visual system
Brainstorming	Process cycle efficiency	Chi-square	Mistake proofing	Project commissioning
Affinity diagramming	Process sizing	T-tests	Setup reduction	Project replicating
Murphy's analysis	Process capability C_p & C_{pk}	Hypothesis testing	Generic pull	PDCA cycle
Surveys	QFD	Queuing theory	Process mapping	5S discipline
Customer requirements trees	Flow down	Analytical batch sizing	5S improvement	Review with sponsor
Nonvalue-added analysis	ANOVA	x-y map	Kaizen	
VOC and Kano analysis	Review with sponsor	Spaghetti diagrams	Poka-yoke	Acceptance sampling
QFD	Process observation	VA/NVA analysis	FMEA	TAKT time
RACI and Quad charts		Takt time /Cycle time	Solution selection matrix	
Stakeholder analysis		Time value chart	Piloting and simulation	
Key process output variabilities KPOV's		5S analysis	Control plans	
Review with sponsor		Review with sponsor	Review with sponsor	

2.2.6 Acceptance Sampling

Data analysis, as explained earlier, is a vital tool in the latest modern management principle that helps establish a baseline data that LSS uses to draw conclusions and improve processes. In most cases, the first steps in the sampling process are to ascertain how much data to collect to be able to effectively evaluate and validate conclusions statistically. In the process of ascertaining data, sample size plays a critical role in acceptance sampling (Franchetti, 2015). One of the common sampling techniques known as acceptance sampling can be defined as a quality control process that entails randomly selecting several items in a large lot to judge whether the entire lot meets the product specification and required quality standards. Based on the number of defective items in a sample, a decision may be made to either to (1) accept the batch, (2) reject the batch and (3) continue sampling until a decision is made (Fallahnezhad, Babadi, Momeni, Sayani, & Akhoondi, 2015). Acceptance sampling can be applied in many business areas. For example, in the manufacturing sector, acceptance sampling is used as an inspection tool to determine whether incoming materials or parts meet the quality requirements of the finished product to be manufactured. Acceptance sampling can also be used to inspect whether the unfinished units are within the acceptable quality level to move to the next phase in the production line, or to ensure that the quality of finished products at the end of the production line is within the customer's acceptable quality level.

There are two main categories of acceptance sampling plans based on the quality characteristics of the units to be inspected, namely, acceptance by variables and acceptance by attributes. Acceptance by variables deals with sampling based on quality characteristics that are measured on a numerical scale like measuring the mean, standard deviation, and the variability of the produced unit or process (temperature, diminution, etc.). Thus, Acceptance by variables leads to a continuous measurement. On the other hand, acceptance by attributes is solely based on quality characteristics expressed in the form of a "go-no-go" and "bad or good" basis.

Of the two types, acceptance sampling by attributes is the most commonly used form of acceptance sampling plans (Borror, 2009). Sampling based on attributes is also easy to implement and more cost effective than sampling based on variables (Aslam, Muhammad; Khan, Nasrullah; and Khan, 2015). The Department of Defense (DOD) of the United States has studied and implemented acceptance sampling since 1989. Two widely use sampling standards published by the DOD include the MIL-STD-105E (military standard-105E), and the MIL-STD-1916. Those two widely used standards will be discussed later in part 2.2.6.4. In this dissertation, the acceptance sampling by attributes will be emphasized. Figure 10 shows the process of acceptance sampling.

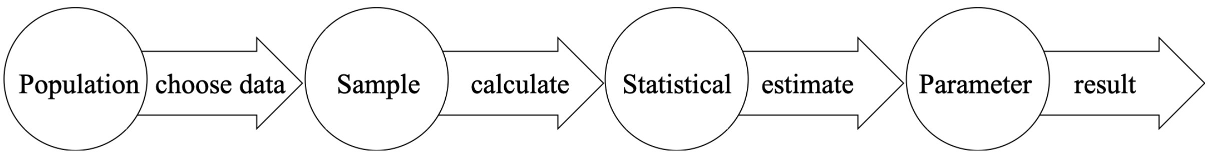


Figure 10: Acceptance Sampling Process

2.2.6.1 Designing of Acceptance Sampling Plans

In the process of designing any of the acceptance sampling plans, two levels of quality are usually considered . These two quality levels are known as the acceptable quality level (AQL), and the lot tolerance proportion defective (LTPD)/rejectable quality level (RQL). AQL is known to be the consumers' preferable quality level choice and it is usually incorporated into a purchase order or contract. According to the American Society for Quality (ASQ), AQL is defined as "the quality level that is the worst tolerable process average when a continuing series of lots is submitted for acceptance sampling" (Borror, 2009). The producer of the product often tries to achieve AQL, which is associated with producer risk (α) . Producer's risk or alpha risk is characterized by rejecting a good lot as a result of the nature of the sampling technique (Walker, Elshennawy, Gupta, & McShane Vaughn, 2012).

LTPD/RQL, which is the second level of quality, refers to the poorest quality in a specific lot that should be accepted. It is usually linked to the worst level of quality the consumer can tolerate but wish to reject because it meets the minimum acceptance quality requirement. It is associated with what is referred to as consumer risk (β). Consumer risk or beta risk is simply defined as the risk associated with accepting a poor lot based on the nature of the sampling technique (Walker et al., 2012). That being said, RQL is the defective rate at which there is β probability that produced items will be accepted by the consumers.

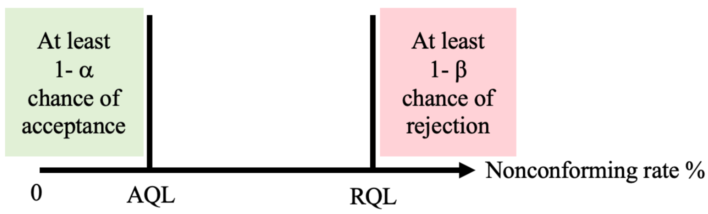


Figure 11: Acceptance Quality Level and Rejectable Quality Level

2.2.6.2 Advantages and Disadvantages of Acceptance Sampling by Attribute

Acceptance sampling can be seen as a reactive quality control process and a method for inspecting not only a manufacturing product but also service (Allen, 2010). It also can be considered as an auditing technique and has its benefits and flaws. One of the key merits of acceptance sampling by attribute is its economic benefits. For example, if the cost of inspection is high and or the time to inspect a product is long, sampling by attribute can be preferable as opposed to 100 percent inspection. Additionally, acceptance sampling also reduces the risk of damaging a product due to process handling. Acceptance sampling also helps to reduce inspection errors significantly in the sense that repetitive inspection such as 100 percent inspection can lead to inspector fatigue, thereby preventing the inspector from identifying all nonconforming products.

Acceptance sampling also has certain key demerits that may hinder its implementation. One of the major demerits of acceptance sampling is the risk associated with rejecting "good" lots or accepting "poor" lots identified as the producer's risk and the consumer's risk, respectively (Mitra, 2016).

2.2.6.3 Types of Attribute Acceptance Sampling Plans

As discussed earlier, one of the key classifications of sampling plans is associated with attributes. This part of the paper will briefly discuss acceptance sampling plans for attributes with their own advantages and disadvantages. These attribute-based acceptance sampling plans include those with a fixed sample size such as single, double, and multiple acceptance sampling, as well as plans with variable sample sizes, referred to as sequential sampling plans. The implementation of any of these types of sampling plans is based on several factors that include ease of usage, the general quality of incoming lots of product from the suppliers, that is, are the incoming parts (material) of excellent quality history or questionable quality history?

2.2.6.3.1 Single Sampling Plan

In a single sampling plan, a lot-sentencing technique is used. In this case information retrieved from one sample is used to make a decision to either accept or reject a particular lot of size *N.* The two main features of this type of sampling plan are the samples size denoted by *(n)* and the acceptance number denoted by *(c),* which can randomly be selected

(Montgomery, 2013). In this case, the acceptance number *(c)* is defined as the acceptable number of defective items allowed in a lot *(N)* to be considered as an acceptable lot and pass the inspection test successfully.

One of the key merits of a single sampling plan is related to its ease of implementation. Moreover, the fixed or rigid nature of its sample size makes it difficult to take advantage of the potential cost saving of reduced inspection when incoming parts are either good or bad (Borror, 2009). The graph below shows how a single sampling plan is used. It starts with defining the number sample size *(n)* needed to be inspected from the current lot and the acceptance number *(c)* for the lot. The sample size *(n)* is inspected in order to detect any nonconforming items or defectives items (d_1). The number of nonconforming items (d_1) that is found in the sample size is compared with the acceptance number of the lot that was determined before. If the detected nonconforming items (d_1) are more than the acceptance number, the lot is rejected. On the contrary, if the number of nonconforming items is less than or equal to the acceptance number, then the lot is accepted and termed to be good (Montgomery, 2013), as shown in Figure 12.

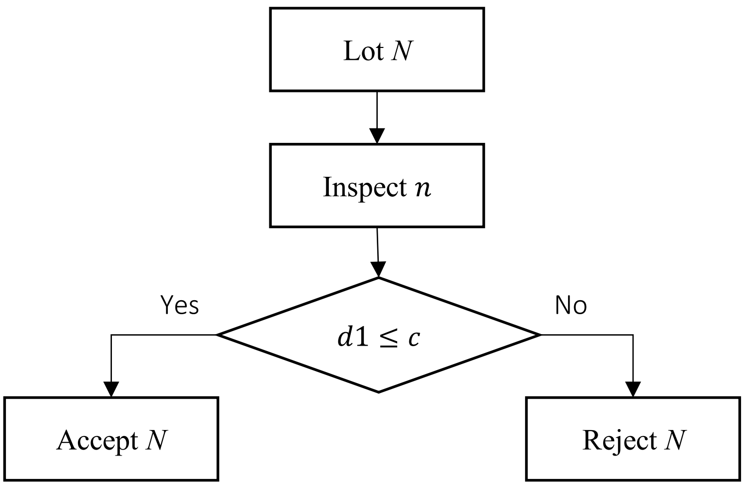


Figure 12: Single Acceptance Sampling Plan

2.2.6.3.2 Double Sampling Plan

As the name implies, with a double sampling plan, a second sample is necessary before a lot can be deemed good or bad. In other words, the double sampling particularly works when the result of the first inspected sample fails to show clear evidence whether to accept or reject the lot (Allen, 2010). Unlike single sampling, a double sampling plan can be characterized by six variables: the size of the first sample (n_1), the acceptance number of the first sample (c_1), the size of the second sample (n_2), the cumulative acceptance number of both the first and second samples (c_2), the nonconforming units (d_1 *and* d_2), the rejection number of the first sample (r_1), and the rejection number of the second sample (r_2).

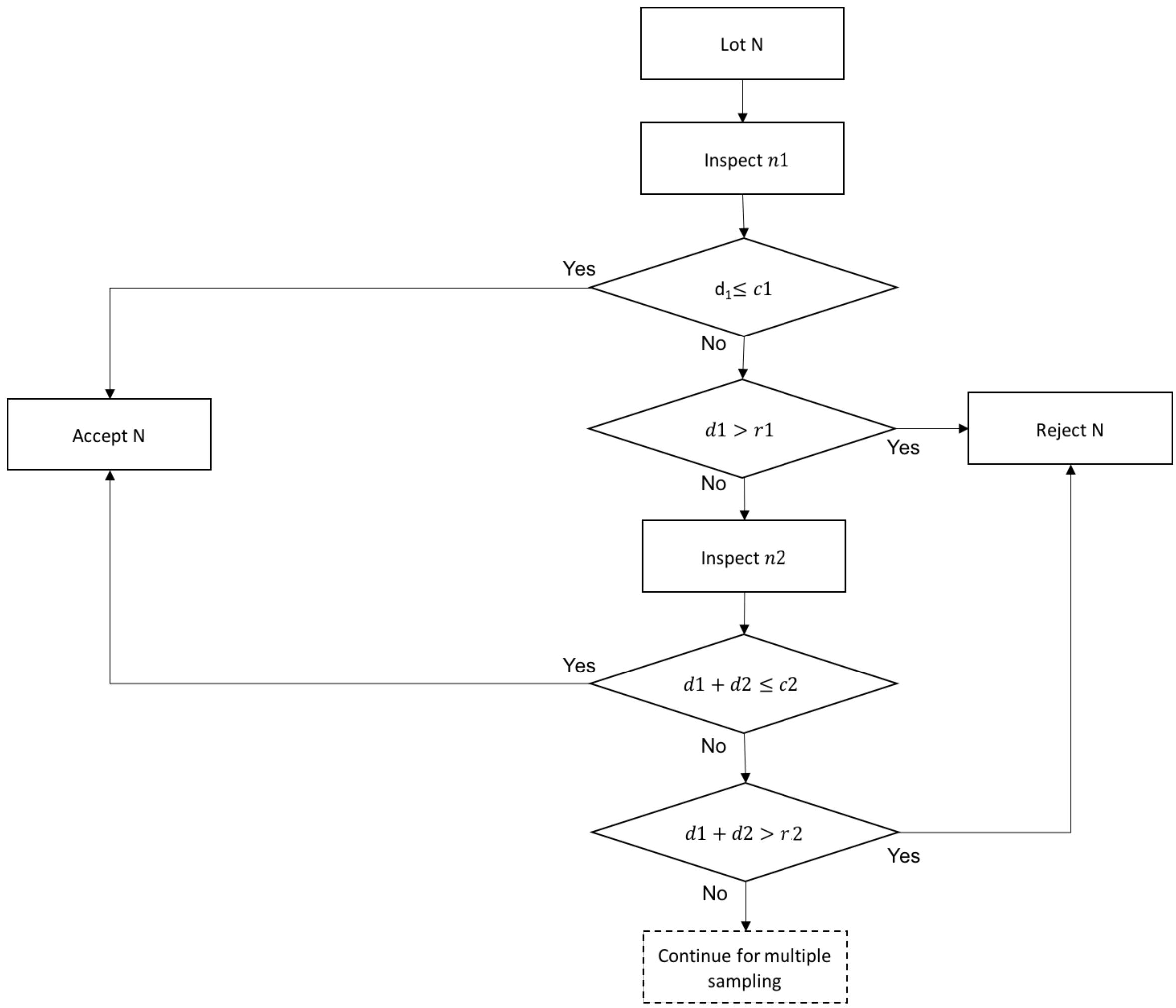


Figure 13: Double and Multiple Acceptance Sampling Plans

Figure 13 shows the process of accepting and rejecting the lot (N) in a double sampling plan. From Figure 13, a lot can be accepted or rejected from the first sample without necessarily taking a second sample. That is to say that the lot can be accepted based on the first sample only when the nonconforming units are less than or equal to the acceptance number for the first sample $d_1 \leq c_1$. The lot can be rejected when the number of nonconforming units is greater than the rejection number $d_1 > r_1$. However, if the number of nonconforming units is greater than the acceptance number of the first sample but less than the rejection number of the first sample, then the second sample n_2 must be drawn. If the number of nonconforming units from the first sample plus the nonconforming units from the second sample are less than or equal to the acceptance number of the second sample $d_1+d_2 \leq c_2$, the lot will be accepted. If not, lot N is rejected (Schilling & Neubauer, 2017).

2.2.6.3.3 Multiple Sampling Plan

With respect to a multiple sampling plan, more than two samples are required to make a decision to either reject or accept a lot. It is simply an extension of the double sampling plan but with an extension of the number of samples to be performed. See Figure 13. According to ANSI/ASQ Z1.4-2003, a multiple sampling plan can be performed up to seven times (Allen, 2010; Borror, 2009). The main advantage of a multiple sampling plan is the fact that smaller samples are required at each stage compared with single and double sampling plans. This advantage makes it more economically efficient than the single and double sampling (Schilling & Neubauer, 2017).

2.2.6.3.4 Sequential Sampling

In this section of the paper, a more specific sampling plan that is used in special sampling cases is considered. Sequential sampling plans aim to reduce the inspection time and effort, to simplify, or to provide better protection and agility under unique conditions (Mitra, 2016). Unlike the fixed plans (single, double, multiple), this plan requires a smaller number of inspected units, with the predicted sample size adopting to the estimated rate of defectives. For that reason, this plan can be beneficial to implement especially when testing is costly and or the units needing to be inspected are fragile and easy to break (Childs & Chen, 2011). The mechanism behind sequential sampling is similar to multiple sampling in the sense that the number of items required for sampling is dictated by the results of the sampling process itself, hence dynamic and adaptive. In sequential sampling, a decision to accept a lot, reject a lot or continue with the sampling process is solely based on the outcome of the cumulative inspection results. Sequential sampling can be used to inspect products until 100% inspection is reached but this might undermine the core principle of sequential sampling, which is to reduce inspection time and sample sizes. For this reason, the rule of thumb is that sequential sampling cannot exceed three times the number to be inspected by an equivalent single sampling plan, at which point a decision must be made to terminate the plan and notify the vendor to demonstrate an improved product quality before any further product can be accepted (Mitra, 2016).

Sequential sampling can be categorized based on the sample size. If the sample size chosen at each stage is greater than one, the process is referred to as group sequential sampling. On the other hand, if the sample size chosen for inspection at each stage is one, then the process is referred to as item-by-item sequential sampling (Montgomery, 2013). Often times, item-by-item sequential sampling is used especially in cases where it is necessary to reach a decision to either accept or reject a lot as soon as possible to help reduce inspection cost and time.

The item-by-item sequential sampling plan is based on the sequential probability ratio test (SPRT) developed by Wald (1947). The primary objective of SPRT is to test a lot or a batch of items based on optimal sample size in order to make a decision to accept or reject the lot (Chetouani, 2013; Starvaggi, 2014). It is considered to be one of the most commonly used methods worldwide especially in manufacturing. SPRT is used to test a hypothesis regarding the proportion of non-conforming items in a lot or the quality level of the current lot.

$$Ho: p \leq p_1$$

$$H_a: p \geq p_2$$

Where p is the quality parameter of a lot under inspection, p_1 is the acceptable quality level of nonconformities, and p_2 is the rejection quality level of nonconforming units or items.

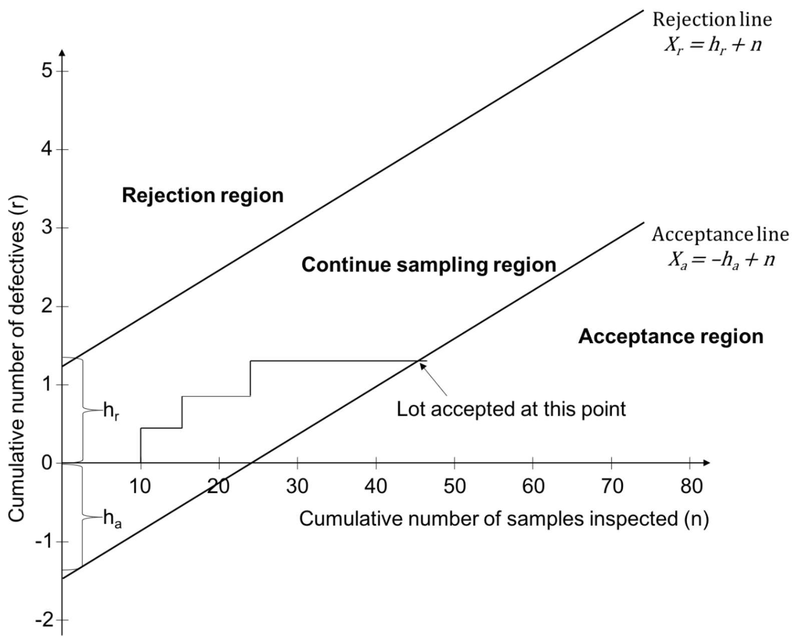


Figure 14: Item-by-item Sequential Sampling Plan

Figure 14 depicts the operation of item-by-item sequential sampling. The cumulative number of defectives (r) is plotted against the cumulative number of samples/units inspected (*n*). There are two main lines depicting acceptance and rejection line (X_a, X_r). These lines, as the name implies, determine the regions for acceptance and rejection and in between these two lines is the region that gives room to continue sampling until an inspection decision is reached. When a sample point falls on or below the acceptance line, a decision is made to accept the lot. On the other hand, if the sample point falls on or above the rejection line, a decision is automatically made to reject the whole lot. These two lines are influenced by the predetermined parameters from above (α, P_1, β, P_2) and are generated using defined equations (Mitra, 2016; Montgomery, 2013).

$$X_a = -h_a + sn \text{ (acceptance line)}$$

$$X_r = h_r + sn \text{ (rejection line)}$$

These equations are dictated by the producer's risk (α) and its associated acceptable quality level (P_1) and the consumer's risk (β) and its associated rejectable quality level (P_2). The equations are also controlled by ($-h_a$, h_r), which is the intercept on the vertical axis, *s* is the slope, and k is the average acceptable number of defective unites. The equations below show how k, h_a, h_r and the s are calculated (Mitra, 2016; Montgomery, 2013).

$$k = \log\frac{p_2(1-p_1)}{p_1(1-p_2)}$$

$$h_a = \frac{\left(\log\frac{1-\alpha}{\beta}\right)}{k}$$

$$h_r = \frac{\left(\log\frac{1-\beta}{\alpha}\right)}{k}$$

$$s = \frac{log\frac{1-p_1}{1-p_2}}{k}$$

Given the agility requirement for sequential sampling, the goal of this research is to develop a novel sequential sampling framework, the reinforcement learning methodology, which is introduced in Section 2.4.

2.2.6.4 MIL-STD-105E and MIL-STD196

Two well-known acceptance sampling plans standards for dealing with attribute data are military standard-105E (MIL-STD-105E), and military standard-1916 (MIL-STD-1916). The United States Department of Defense (DOD) published the first standard in 1989 during World War II to provide procedures and reference tables for designing and implementing the appropriate methods for acceptance sampling by attributes. MIL-STD-105E, also known as ANSI Z1.4 and ISO 2859, is one of the most widely used systems in acceptance sampling worldwide. MIL-STD-105E is considered to be the traditional version of the U.S. DOD for fixed sampling plans where the acceptable quality level (AQL) and the producer's risk is the primary focal of this standard (MIL-STD-105E, 1989). In other words, when implementing MIL-STD-105E for inspection, the lot size, inspection level, AQL level, type of inspection, and type of sampling plan must be defined.

In 1996 the U.S. DOD adopted MIL-STD-1916 to be the new version of accepting sampling system to replace the traditional MIL-STD-105E sampling plans (MIL-HDBK-1916, 1999). Unlike MIL-STD-105E, MIL-STD-1916 is known to be less complicated since it contains fewer tables and less information to use (M. H. C. Li & Tsao, 2011). In contrast, MIL-STD-1916 mainly focuses on "zero accept one rejects" as criteria of whether to accept or reject a batch or lot. If a sample has at least one nonconformance, then the whole batch or lot must be rejected. Using MIL-STD-1916 indirectly forces the producers to have a strict quality control system (Hamzic, 2013). For that reason, MIL-STD-1916 can be used when the contractor or the supplier is extremely confident in the quality of its products. MIL-STD-105E takes into account the three attribute sampling plans (single, double, and multiple). Unlike MIL-STD-105E, MIL-STD-1916 deals with attribute sampling, variable sampling, and continues sampling. Moreover, the only information that needs to be specified in implementing MIL-

STD-1916 is the lot size, type of inspection, and inspection level (MIL-STD-105E, 1989; MIL-STD-1916, 1996)

2.2.6.5 Inspection levels

MIL-STD-105E and MIL-STD-1916 show three types of inspection (severity inspection) during sampling, including normal inspection, tightened inspection, and reduced inspection. The normal severity inspection is the most common type used during the process of inspection, and it can be implemented when the supplier is confident that the quality level meets the acceptable level, whereas, tightened inspection is implemented when a large sample size is required to be inspected because of a history of poor quality, or when the product is newly developed. On the other hand, reduced inspection is implemented when it is a requirement to inspect a small sample size due to a short time or budget or when a company has a reliable quality management system (MIL-STD-105E, 1989; MIL-STD-1916, 1996).

Seven verification levels (VL) of sampling inspection are proposed in both standards for each sampling plan with their code letters to determine the relationship between the lot or batch size and the sample size. According to MIL-STD-1916, "Verification Level (VL) prescribes the level of significance or utility of a characteristic (attribute) to the user" (MIL-STD-105E, 1989; MIL-STD-1916, 1996).

Table 5 and Table 7 present the verification level for MIL-STD-1916 and MIL-STD105E, respectively. Level I is implemented when it is a requirement to inspect a small sample size due to a short time or budget or when a company has a reliable quality management system. Level II is the normal severity inspection level, and according to MIL-STD-105E, it is the most common level used during the inspection. Level III can be used when a large sample size is required to be inspected because of a history of poor quality or a newly developed item. In contrast, applying level IV-VII requires more effort in sampling, since they can be implemented where “ small sample sizes are necessary for inspection and large sampling risks can or must be tolerated” (ISO 2859-standard).

After determining the appropriate verification level and the lot or batch size, MIL-STD-1916 shown in Table 5 helps to find the proper sample size code letter to carry over to the chosen sampling plan table. Then, the accurate inspected sample size for that particular batch size can be found in the sampling plan standard shown in Table 6.

Table 5: Sample size code letters (MIL-STD-1916, 1996)

Lot or production interval size	Verification levels						
	VII	VI	V	IV	III	II	I
2–170	A	A	A	A	A	A	A
171–288	A	A	A	A	A	A	B
289–544	A	A	A	A	A	B	C
545–960	A	A	A	A	B	C	D
961–1632	A	A	A	B	C	D	E
1633–3072	A	A	B	C	D	E	E
3073–5440	A	B	C	D	E	E	E
5441–9216	B	C	D	E	E	E	E
9217–17408	C	D	E	E	E	E	E
17409–30720	D	E	E	E	E	E	E
30721 and larger	E	E	E	E	E	E	E

Table 6: Attribute Sampling Plans (MIL-STD-1916, 1996)

Code letter	Verification levels								
	T	VII	VI	V	IV	III	II	I	R
	Sample size (n_a)								
A	3072	1280	512	192	80	32	12	5	3
B	4096	1536	640	256	96	40	16	6	3
C	5120	2048	768	320	128	48	20	8	3
D	6144	2560	1024	384	160	64	24	10	4
E	8192	3072	1280	512	192	80	32	12	5

NOTES:

(1) When the lot size is less than or equal to the sample size, 100 percent attributes inspection is required.

(2) One verification level (VL) to the left/right of the specified normal VL is the respective tightened/reduced plan. Tightened inspection of VL-VII is T, reduced inspection of VL-I is R.

When creating a sampling plan using MIL_STD-105E, the information presented in Table 7 helps to find the proper sample size code letter to carry over to the chosen sampling plan table. After locating the sample size code letter and specifying the acceptable quality level (AQL) which is the nonconformities per 100 items by the authority responsible for sampling, then the accurate inspected sample size and the acceptable and rejectable number for that particular batch size can be found in the sampling plan standard (Table 8, Table 9, Table 10).

Example 1

Here is an example of how to perform MIL-STD-105E for single sampling. Suppose a product X is submitted in batches of size N = 500. The buyer and the producer have agreed to the acceptable quality level (AQL) that equal to 1.5% for a normal single sampling plan and using the general inspection level II. The buyer and the producer are looking to find out how many samples are required for inspection, and how many nonconforming items are allowed in order to accept or reject the batch. In this matter, first we find the row that corresponds to the total batch size of 500 from Table 7, so the appropriate row is the 281 to 500. Second, determine the applicable inspection level and find the corresponding column in Table 7. Level II, which stands for standard severity inspection, will be used in this example. So the correct sample size code letter to carry over to Table 8 for a single sample plans will be the letter H. As a result, Table 8 shows that the sample size code H requires a sample size of 50 for inspection from each batch. Since the AQL is 1.5%, Table 8 gives a maximum number of 2 defective items for accepting a single batch and a minimum of 3 defective items for rejecting the batch.

Example 2

Following the same example from the single sampling section, where the batch size is 500, and the AQL is equal to 1.5%, for the double sampling, it can be performed twice if the number of nonconforming units from the first sample size is between the rejectable and acceptable numbers. Looking back to the example, since we have a batch size of 500, the appropriate sample size code letter from Table 7 is H, and the necessary sample size for the first and second sample size is 80 for each one. In this case, Table 9 shows each sample size has a different number of defects as a limit, where the acceptable and rejectable number of defectives for the first sample size are 0 and 3 and for the second sample size are 3 and 4, respectively. So, if the number of inspected items from the first sample size is excellent, meaning that there are 3 defects or fewer, the batch is accepted. If it is terrible, meaning that there are 4 defects or more, then the batch is rejected and there is no need to inspect the second sample. However, if the number of defectives found in the first sample size is 1, then the next sample size of 32 should be inspected with a limitation of an acceptance number of 3 defects and a rejection number of 4 cumulatively. So, if the total number of nonconforming items from the first and second sample size is equal to 3 or less, the batch is accepted, and if it is equal to 4 or greater, it is rejected.

Table 7: Sample Size Code Letters (MIL-STD 105E)

Lot or batch size			Special inspection levels				General inspection levels		
			S-1	S-2	S-3	S-4	I	II	III
2	To	8	A	A	A	A	A	A	B
9	To	15	A	A	A	A	A	B	C
16	To	25	A	A	B	B	B	C	D
26	To	50	A	B	B	C	C	D	E
51	To	90	B	B	C	D	D	F	G
91	To	150	B	B	C	D	D	F	G
151	To	280	B	C	D	E	E	G	H
281	To	500	B	C	D	E	F	H	J
501	To	1200	C	C	E	F	G	J	H
1201	To	3200	C	D	E	G	H	K	L
3201	To	10000	C	D	F	G	J	L	M
10001	To	35000	C	D	F	H	K	M	N
35001	To	150000	D	E	G	J	L	N	P
150001	To	500000	D	E	G	J	M	P	Q
500000	And	Over	D	E	H	K	N	Q	R

Table 8: Single Sampling Plans for Normal Inspection (Master Table)

Sample size Code Letters	Sample size	Acceptable Quality Levels (Normal Inspection)																									
		0.010	0.015	0.025	0.040	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000
		Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re
A	2														↓	0 1		↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	30 31
B	3													↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	30 31	44 65
C	5												↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	30 31	44 65	↑
D	8											↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	30 31	44 65	↑	
E	13										↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	30 31	44 65	↑		
F	20									↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑	↑	↑			
G	32								↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑						
H	50							↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑							
J	80						↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑								
K	125					↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑									
L	200			↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑										
M	315				0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑											
N	500	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑												
P	800		0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑													
Q	1250	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑														
R	2000	↑		1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑															

↓ = Use first sampling plan below arrow. If samples size equals or exceeds lot or batch size, do 100 inspections
↑ = Use first sampling plan above arrow.
Ac = Acceptance number
Re = Rejection number

Table 9: Double Sampling Plans for Normal Inspection (Master Table)

Sample Size Code Letter	Sample	Sample Size	Cumulative sample size	Acceptable Quality Levels (Normal Inspection)†																									
				0.010	0.015	0.025	0.040	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000
				Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re
A				↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	•	↓	↓	•	•	•	•	•	•	•	•	•
B	First	2	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	17 22	25 31
	Second	2	4																	1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27	37 38	56 57
C	First	3	3	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	17 22	25 31	↑
	Second	3	6																1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27	37 38	56 57	
D	First	5	5	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	17 22	25 31	↑	↑
	Second	5	10															1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27	37 38	56 57		
E	First	8	8	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	17 22	25 31	↑	↑	↑
	Second	8	16														1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27	37 38	56 57			
F	First	13	13	↓	↓	↓	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑
	Second	13	26													1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27						
G	First	20	20	↓	↓	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑
	Second	20	40												1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27							
H	First	32	32	↓	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑
	Second	32	64											1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27								
J	First	50	50	↓	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	50	100										1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27									
K	First	80	80	↓	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	80	160									1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27										
L	First	125	125	↓	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	125	250								1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27											
M	First	200	200	↓	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	200	400							1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27												
N	First	315	315	↓	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	315	630						1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27													
P	First	500	500	↓	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	500	1000					1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27														
Q	First	800	800	•	↑	↓	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	800	1600				1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27															
R	First	1250	1250	↑	↑	0 2	0 3	1 4	2 5	3 7	5 9	7 11	11 16	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	1250	2500			1 2	3 4	4 5	6 7	8 9	12 13	18 19	26 27																

↓ = Use first sampling plan below arrow. If samples size equals or exceeds lot or batch size, do 100 inspections

↑ = Use first sampling plan above arrow.

Ac = Acceptance number

Re = Rejection number

• = Use corresponding single sampling plan (or alternatively, use double sampling plan, where available).

Table 10: Multiple Sampling Plans for Normal Inspection (Master Table)

Acceptable Quality Levels (Normal Inspection)†

Sample Size Code Letter	Sample	Sample Size	Cumulative sample size	0.010	0.015	0.025	0.040	0.065	0.10	0.15	0.25	0.40	0.65	1.0
				Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re
A B C				↓	↓	↓	↓	↓	•	↑	↓	∞ 2	∞ 2	∞ 3
												∞ 2	0 3	0 3
												0 2	0 3	1 4
												0 3	1 4	2 5
												1 3	2 4	3 6
												1 3	3 5	4 6
												2 3	4 5	6 7
D	First	2	2	↓	↓	↓	↓	•	↑	↓	∞ 2	∞ 2	∞ 3	∞ 4
	Second	2	4								∞ 2	0 3	0 3	1 5
	Third	2	6								0 2	0 3	1 4	2 6
	Fourth	2	8								0 3	1 4	2 5	3 7
	Fifth	2	10								1 3	2 4	3 6	5 8
	Sixth	2	12								1 3	3 5	4 6	7 9
	Seventh	2	14								2 3	4 5	6 7	9 10
E	First	3	3	↓	↓	↓	•	↑	↓	∞ 2	∞ 2	∞ 3	∞ 4	0 4
	Second	3	6							∞ 2	0 3	0 3	1 5	1 6
	Third	3	9							0 2	0 3	1 4	2 6	3 8
	Fourth	3	12							0 3	1 4	2 5	3 7	5 10
	Fifth	3	15							1 3	2 4	3 6	5 8	7 11
	Sixth	3	18							1 3	3 5	4 6	7 9	10 12
	Seventh	3	21							2 3	4 5	6 7	9 10	13 14
F	First	5	5	↓	↓	•	↑	↓	∞ 2	∞ 2	∞ 3	∞ 4	0 4	0 5
	Second	5	10						∞ 2	0 3	0 3	1 5	1 6	3 8
	Third	5	15						0 2	0 3	1 4	2 6	3 8	6 10
	Fourth	5	20						0 3	1 4	2 5	3 7	5 10	8 13
	Fifth	5	25						1 3	2 4	3 6	5 8	7 11	11 15
	Sixth	5	30						1 3	3 5	4 6	7 9	10 12	14 17
	Seventh	5	35						2 3	4 5	6 7	9 10	13 14	18 19
G	First	8	8	↓	•	↑	↓	∞ 2	∞ 2	∞ 3	∞ 4	0 4	0 5	1 7
	Second	8	16					∞ 2	0 3	0 3	1 5	1 6	3 8	4 10
	Third	8	24					0 2	0 3	1 4	2 6	3 8	6 10	8 13
	Fourth	8	32					0 3	1 4	2 5	3 7	5 10	8 13	12 17
	Fifth	8	40					1 3	2 4	3 6	5 8	7 11	11 15	17 20
	Sixth	8	48					1 3	3 5	4 6	7 9	10 12	14 17	21 23
	Seventh	8	56					2 3	4 5	6 7	9 10	13 14	18 19	25 26
H	First	13	13	•	↑	↓	∞ 2	∞ 2	∞ 3	∞ 4	0 4	0 5	1 7	0 2
	Second	13	26				∞ 2	0 3	0 3	1 5	1 6	3 8	4 10	7 14
	Third	13	39				0 2	0 3	1 4	2 6	3 8	6 10	8 13	13 19
	Fourth	13	52				0 3	1 4	2 5	3 7	5 10	8 13	12 17	19 25
	Fifth	13	65				1 3	2 4	3 6	5 8	7 11	11 15	17 20	25 29
	Sixth	13	78				1 3	3 5	4 6	7 9	10 12	14 17	21 23	31 33
	Seventh	13	91				2 3	4 5	6 7	9 10	13 14	18 19	25 26	37 38
J	First	20	20	↑	↑	∞ 2	∞ 2	∞ 3	∞ 4	0 4	0 5	1 7	0 2	↑
	Second	20	40			∞ 2	0 3	0 3	1 5	1 6	3 8	4 10	7 14	
	Third	20	60			0 2	0 3	1 4	2 6	3 8	6 10	8 13	13 19	
	Fourth	20	80			0 3	1 4	2 5	3 7	5 10	8 13	12 17	19 25	
	Fifth	20	100			1 3	2 4	3 6	5 8	7 11	11 15	17 20	25 29	
	Sixth	20	120			1 3	3 5	4 6	7 9	10 12	14 17	21 23	31 33	
	Seventh	20	140			2 3	4 5	6 7	9 10	13 14	18 19	25 26	37 38	

Sample Size Code Letter	Sample	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000
		Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re
A B C		∞ 4	0 4	0 5	1 7	0 2	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
		1 5	1 6	3 8	4 10	7 14										
		2 6	3 8	6 10	8 13	13 19										
		3 7	5 10	8 13	12 17	19 25										
		5 8	7 11	11 15	17 20	25 29										
		7 9	10 12	14 17	21 23	31 33										
		9 10	13 14	18 19	25 26	37 38										
D	First	0 4	0 5	1 7	0 2	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	1 6	3 8	4 10	7 14											
	Third	3 8	6 10	8 13	13 19											
	Fourth	5 10	8 13	12 17	19 25											
	Fifth	7 11	11 15	17 20	25 29											
	Sixth	10 12	14 17	21 23	31 33											
	Seventh	13 14	18 19	25 26	37 38											
E	First	0 5	1 7	0 2	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	3 8	4 10	7 14												
	Third	6 10	8 13	13 19												
	Fourth	8 13	12 17	19 25												
	Fifth	11 15	17 20	25 29												
	Sixth	14 17	21 23	31 33												
	Seventh	18 19	25 26	37 38												
F	First	1 7	0 2	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	4 10	7 14													
	Third	8 13	13 19													
	Fourth	12 17	19 25													
	Fifth	17 20	25 29													
	Sixth	21 23	31 33													
	Seventh	25 26	37 38													
G	First	0 2	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	Second	7 14														
	Third	13 19														
	Fourth	19 25														
	Fifth	25 29														
	Sixth	31 33														
	Seventh	37 38														
H	First	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
J	First	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑

↓ = Use first sampling plan below arrow. If samples size equals or exceeds lot or batch size, do 100 inspections
↑ = Use first sampling plan above arrow.
Ac = Acceptance number
Re = Rejection number
• = Use corresponding single sampling plan (or alternatively, use double sampling plan, where available).
† = If, after the second sample, the acceptable number has been exceeded, but the rejection number has not been reached, accept the lot, but reinstate normal inspection.
∞ = acceptance not permitted at this sample size

2.2.6.5.1 Acceptance Sampling Performance

To measure the performance of each acceptance sampling plan, the ***operating characteristics (OC) curve*** measurement has to be implemented. This curve plots the probability of accepting the current lot for a range of proportion of defective products. The outputs or results of the OC curve show how well a sampling plan differentiates between good and bad lots and help to determine which acceptance sampling plan is more appropriate for a production inspection process. It is also noted that there are other measurement techniques to test the performance of a sampling plan such as, but not limited to, the average quality level, the average sample number (ASN), the average quality level of the outgoing items, the average total inspection of items (ATI) and many other techniques (Mitra, 2016; Schilling & Neubauer, 2017).

2.3 Machine Learning and Reinforcement Learning

2.3.1 Introduction

This part of the dissertation gives a general idea of reinforcement learning (RL) and the model that is related to the proposed research. The introduction of RL will include the following: the background of RL, the history of RL, the components of RL and its mechanism, and the special cases of RL. In recent decades, machine learning has become an effective subtopic of artificial intelligence and it is geared towards the development of complex artificial learning systems (Mitchell et al., 2018; Si, Barto, Powell, & Wunsch, 2012).

Currently, the manufacturing industry has been embracing the fourth industrial revolution's technologies such as machine learning to meet the quality requirements of a product and optimize production processes. Machine learning (ML) is known to be an automated learning process that observes data from the past and analyzes them to make predictions, classification, clustering patterns, and actions. ML algorithms can be divided into four categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (RL) (Lee, Shin, & Realff, 2018). ML is a supervised learning whereby the features in the data under study are labeled or targeted (labeled training examples). The supervised learning process helps predict the relationship between the labeled examples for classification and for better decision making (Kulkarni, 2012). However, if the features are available but not labeled (unlabeled training examples), then it is unsupervised machine learning. The unlabeled features can be used on multiple tasks such as clustering, comparison, and feature extraction based on their distribution. Machine learning can also be semi-supervised learning when a majority of the training examples are unlabeled, and some are labeled. Semi-supervised learning uses the labeled examples and the unlabeled examples combined to predict the probability distribution and find the similarity between labeled and unlabeled examples. Lastly, reinforcement learning is different from the other types of machine learning, since it is a method of self-learning and acting based on observed data (Lee et al., 2018).

For the past 40 years, researchers have studied mainly supervised, semi-supervised and unsupervised learning and have been able to discuss the importance of their usage, their downside and what can be done to improve their application. However, in recent years, RL has surfaced and its application has been seen in many fields of study (Zarandi, M., Moosavi, H., & Zarinbal, 2013). This dissertation focuses on the application of RL in manufacturing, and in particular as a way to model acceptance sampling.

Reinforcement Learning is known to be one of the most interesting topics in machine learning and sits at the intersection of many professional disciplines, including computer science, engineering, mathematics, economics and psychology, just to mention a few. According to the book, “Artificial Intelligence for All: An Abiding Destination,” RL can be defined as “a learning method for an agent that interacts with its environment by producing actions, thus discovering errors or rewards” (Pathak & Tiwari, 2018). The main reason why RL has the ability to be adoptive when applied in various sectors is that it has the capability to solve complex sequential decision-making problems and to find an optimal course of action (Gatti, 2015). For example, RL can solve problems associated with traffic light control, process scheduling, inventory control, maintenance management, supply chain management and supplier selection (Zarandi, M., Moosavi, H., & Zarinbal, 2013).

2.3.2 The History of RL

The history of reinforcement learning stems from two major roots. The first root is attributed to learning by trial and error, which was used in the psychology of animal learning (Fathi, Maihami, & Moradi, 2013). Some of the earliest work in artificial intelligence used the trial and error method that later helped to revive reinforcement learning in the early 1980s (Sutton & Barto, 2012). The second root, on the other hand, is attributed to the problem of optimal control and its solution, using value functions and dynamic programming. According to the book “Reinforcement learning: An Introduction.”, the term “optimal control” is used to describe the problem of designing a controller to minimize a measure of a dynamical system’s behavior over a period of time. One of the approaches to the “problem of optimal control” was developed in the mid-1950s by Richard Bellman and other scholars by extending a nineteenth century theory of Hamilton and Jacobi (Sutton & Barto, 2012). It is believed that Edward Thorndike is credited with the term trial-and-error learning, which deals with the notion that "actions followed by good or bad outcomes have their tendency to be reselected and altered accordingly." This is called the "Law of Effect" since it describes the effect of reinforcing events on the predisposition to select actions (Sutton & Barto, 2018).

2.3.3 The Mechanism Behind RL

The term Reinforcement Learning basically refers to learning what to do by mapping situations in order to maximize what is termed as a numerical reward signal (Mannion, Devlin, Mason, Duggan, & Howley, 2017). The basic underlying principle of reinforcement learning is that the learner and the decision-maker or the brain of a system, who, in many books and research work is referred to as an *agent,* is not told which actions to take as implied in many forms of machine learning, but must rather find which actions would generate the most reward by trying an action or a combination of actions (H. Li, Cai, Liu, Lin, & Wang, 2018).

The two main features of reinforcement learning are trial-and-error search and delayed reward. As the name implies, trial-and-error search is simply when the agent takes an action or actions and waits for either a positive or negative feedback to be able to make a final decision. A negative feedback from an action taken will alert the agent to take a different action until a positive feedback is achieved. Thus, in a trial-and-error search, the agent keeps trying different actions until a positive signal propels it to move forward with its selected action (Sutton & Barto, 2018). Delayed reward, on the other hand, implies that the agent takes a long sequence of actions, receiving insignificant reinforcement, then finally arrives at a state with high reinforcement. The agent must be able to learn which actions are desirable based on the expected reward or return that can take place arbitrarily far in the future (Sutton & Barto, 2018).

One of the stumbling blocks of reinforcement learning is associated with the trade-off between what is termed as exploration and exploitation. For a reinforcement agent to secure a reward, the agent must select actions that it has tried and executed in the past and concluded to be effective in generating and maximizing an acceptable outcome or reward in the long term (Azizzadenesheli, Lazaric, & Anandkumar, 2017). However, it is important to point out that, in order to discover such a rewarding action, the agent has to experiment with actions that it has not selected before. Thus, the agent has to exploit previous actions to maximize the reward. The agent also has to explore beyond its past actions to achieve a better action selection in the future. From a general perspective, the underlying principle of both exploration and exploitation cannot be pursued without the agent failing at a task. The agent must carryout the concept of trial-and-error on a variety of actions and select those that tend to be effective in addressing a problem (Kulkarni, 2012; Sutton & Barto, 2018).

2.3.4 Agent's History in RL

The history of an agent in RL is defined as a tuple (A, S, R) where A is a sequence of actions $(A_0,\ A_1, A_2\ , A_3, \dots A_t)$, S is a sequence of states $(S_0,\ S_1, S_2\ , S_3, \dots S_t)$, and R is a sequence of rewards $(R_0,\ R_1, R_2\ , R_3, \dots R_t)$ that it has seen so far. So, all of those are the observable variables up to time (t). What happens next depends on those variables.

$$H_t = A_0, S_1, R_1, \dots\dots\dots\dots, A_t, S_t, R_t$$

Where H_t is the agent's history at time t.

However, according to Sutton and Barto in their book, "Reinforcement Learning: An Introduction," the whole history is not very useful because it is going to contain huge data, and this will make it impossible or difficult for the agent to track all past actions within microseconds. In RL, the *history* will be substituted as a state that is the summary of information used to determine what happens next. In other words, in RL we replace the history with a summary of the history that captures all the information that is needed to determine what happens next. Formally, state (S_t) is the function of the history and we can only look at the last observation (Silver, 2015).

$$S_t = f\ (H_t)$$

A state (S_t) is whatever information is accessible to the agent and it follows a Markov process if and only if the probability of the next state ($P[S_{t+1}]$) conditioned on the current state (S_t) is equal to the probability of the next state (S_{t+1}) conditioned on all previous states. In other words, we can ignore all the previous states and just retain the current state (Oddi & Pietrabissa, 2013). Then we will get the same characterization of the future.

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, S_2, \dots \dots \dots, S_t]$$

In other words, we only need to store the current state because the history will not give any additional information about what will happen in the future. Hence, the current state is a sufficient statistic of the future and fully characterizes the distribution of future action (Sutton & Barto, 2018).

2.3.5 Agent and Environment Interaction

The agent in RL is often controlled by an algorithm that is responsible and dictates the action taken by the agent toward a stochastic environment, hence the term agent's state (Gatti, 2015). Figure 15 is a visual representation of how the agent and the environment interact (Gatti, 2015; Sutton & Barto, 2018). The chart shows that at each state or step, the agent undertakes an action based on both the information that was received from the previous state and the accessible actions in the current one (Sutton & Barto, 2018). The next action the agent will make will be based on the result of its action, the agent will receive a new reward, and go to new state.

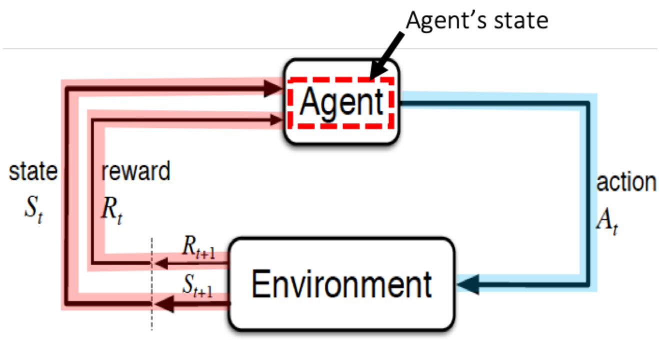


Figure 15: The Agent-Environment Interaction (Sutton & Barto, 2018)

Going into details of the interaction between the agent and the environment happens at each discrete step, $t = 0, 1, 2, 3, \ldots, T$. In RL, the agent makes its sequential decision's action as a function of a signal from the environment called the environment's state, $S_t \epsilon\ S$, and S is the number of possible states the agent can have. The environment state at time t (S_t), is basically defined as the information that is used within the environment to determine what happens next. The agent in this case has no prior knowledge of what is within the environment, so the agent's action will depends on what it receives from the environment's state, $A_t \epsilon\ A(S_t)$, where $A(S_t)$ is the set of a potential actions agent can make in state S_t.

The goal of the agent's actions is to maximize the total amount of reward R_t it receives over a period of time. For the agent to maximize the total future rewards (G_t), the agent must consider an action that maximizes not only the immediate reward in each time step, but also the expected return of the future steps. In the case where the reward is received immediately in each step, then the return will be the sum of the total rewards (Sutton & Barto, 2018).

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} \ldots \ldots + R_T$$

Where R_T is the final step and it is called an episodic task. On the other hand, if the number of actions taken by the agent toward the environment is an infinite *(T=∞)* , then we call it continuing task. In this case, the agent's behavior is to find the optimal policy to maximize the expected discounted return.

The discounted return expresses the value of the future expected reward at the present time, where the value of the reward collected at *k* time steps in the future is worth γ^{k-1}, where γ is the discount factor rate and has to be between zero and one ($0 \leq \gamma \leq 1$) (Sutton & Barto, 2018).

The agent's action A_t depends not only on maximizing the immediate reward but also maximizing the expected reward. For that reason, if $\gamma = 0$, then the agent chooses its action to maximize only the immediate reward in each state and use the sum of the total reward equation.

On the other hand, when $\gamma = 1$, the agent has to choose its action to not only increase the immediate reward but also to increase the future expected reward. That is because receiving an immediate reward can influence the future reward. In this case, the cumulative discounted reward is (Sutton & Barto, 2018).

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \dots \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

In each time step the environment receives an action (A_t) that is taken by the agent and emits an observation (S_{t+1}) as well as a numerical reward ($R_{t+1} \in \mathbb{R}$). Since the agent bases its selection of actions on past experiences and reward, it will select its future actions based on previous observation (S_{t-1}), and rewards (R_{t-1}) (Silver, 2015). At each discrete time step, the agent's selection action is followed by what is called an agent's policy that is represented by π_t. The agent's policy π_t in reinforcement learning can be categorized into two types: deterministic policy and stochastic policy. Deterministic policy can be defined as a function that maps each state to a particular action followed by policy π: $S \rightarrow A$. On the contrary, stochastic policy is a function that maps a specific state $S_t = s$ to probabilities of performing one particular action over another $A_t = a$ at step t.

$$\pi_t(A_t \mid S_t) = P(A_t = a \mid S_t = s)$$

It is not enough for the agent in reinforcement learning to know its policy in selecting a specific action from a specific state (agent's policy) to maximize its expected reward and move to another state see Figure 16 since performing one action over the other available actions in a given state can increase or decrease the expected return.

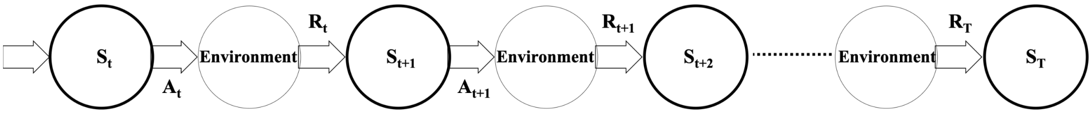


Figure 16: Agent's Transition State Diagram

For that reason, it is important for the agent to estimate in advance how good it is to be in a specific state or how good to perform a specific action in a given state in order to maximize its expected reward. Two value functions are useful to the agent to increase its expected reward with respect to the policy π viz, the state-value function and the action-value function. Both functions are based on the well-known Bellman's equation.

The state value function for policy π, denoted as V_π, gives the agent an insight into how good is the expected return (value of a state) in any given state S_t at any time t and following policy π. In other words, the value function of each state S_t is the total expected future reward given that the agent in state S_t and followed policy π.

$$V_{\pi}(S_t) = \mathbb{E}_{\pi}[G_t | S_t]$$
$$= \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t\right]$$

In like manner, the action-value function for policy π, denoted as Q_{π}, gives the agent insight into how good the expected return is for the agent to perform a particular action from a set of available actions in a given state $A_t \epsilon\ A(S_t)$ following policy π. The function below shows how to calculate the action-value function under policy π:

$$Q_{\pi}(S_t, A_t) = \mathbb{E}_{\pi}[G_t | S_t, A_t]$$
$$= \mathbb{E}_{\pi}\left[\sum_{k=1}^{\infty} \gamma^k R_{t+k+1} | S_t, A_t\right]$$

2.3.6 Optimal policies and Value functions.

Whenever the agent interacts with its environment, there is always one policy better than or equal to the other policies. Thus, the agent keeps interacting with its environment to find an optimal policy, denoted π*, that maximizes the long term expected reward or is at least as equal to all other policies.

$$\pi^* = argmax_a R\ (s, a)$$

The optimal policy is coupled with an optimal state-value function V_{π^*} as well as the optimal action-value function Q_* where the optimal state-value function tells the reinforcement learning agent what the maximum expected return is obtained by being in a certain state *s,* following policy π.

The optimal action-value function, on the other hand, gives the maximum expected return for taking a particular action *a* at state s (state-action pair (s, a)) following policy π.

$$V^*(s) = max_\pi V_\pi \ (S)$$
$$Q^*(s,a) = max_\pi q_\pi \ (s,a)$$

The optimal value-state function known as Bellman's optimality equation for V* provides the maximum value of a state S_t from taking the best action from that state and the reward received R_t from performing that action A_t added to the discounted value from the next state S_{t+1} weighted by the probability transition followed by policy π.

$$V^*(s) = max_a \ \mathbb{E}[\ R_{t+1} + \gamma V_* \ (S_{t+1})|S_t = s, A_t = a]$$
$$= max_{a \in A(s)} \sum_{s'} P(s'|s,a)\ [r(s,a,s') + \gamma V_*(s')]$$

Likewise, the optimal action-value function is the second Bellman's optimality equation that provides the maximum expected discounted return from any given state-action pair (s,a) at any discrete time t. In other words, the Q function gives the maximum expected return (or the quality) of executing A_t at state S_t following policy π.

$$Q^*(s,a) = \mathbb{E}[\ R_{t+1} + \gamma max_{a'} \ q_* \ (S_{t+1}, a')|S_t = s, A_t = a]$$
$$= \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma max_{a'} \ q_*(s',a)]$$

Although the value function V(s), and the quality function Q (s, a) are useful tools to RL in some sense, the agent in reinforcement learning wants to act and learn what action to take in each state; hence, the agent should learn the action-state value function (Q-value), not the state-value function (V(s)). The Q-value function is also beneficial in a situation of RL where an agent does not know the rewards and the transition probability functions in advance. Therefore, the Q-value is calculated by using experienced data without the need of knowing the transition probability or the reward model from the environment.

2.3.7 Q-Learning and Sarsa

The Q-learning algorithm is a major topic for reinforcement learning. It dynamically iterates the action-value function (Q) for each state *s* until the Q-function estimates the optimal action-value function Q^*. The goal of Q-learning is to teach the agent to act optimally by lowering the cost and to find the optimal policy. The Q-learning algorithm starts by giving each action a value; the agent then performs the chosen action and after the algorithm measures the reward received from that action, the value of Q is updated (Sutton & Barto, 2018). This value iteration process is mathematically represented as:

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + a[R(S_t, A_t,) + \gamma \; max_a \; Q(S_{t+1} + a) - Q(S_t, A_t)]$$

Where the first $Q(S_t, A_t)$ is the new Q value, the second $Q(S_t, A_t)$ represents the current Q value, a is the learning rate, $R(S_t, A_t,)$ is the current reward, $max_a \; Q(S_{t+1} + A)$ is the maximum expected future reward given all the available actions A in state S_{t+1} .

On the other hand, Sarsa, which stands for state-action-reward-state, very much resembles Q-learning. The major difference between Sarsa and Q-learning is that Sarsa is a deterministic policy algorithm. It implies that Sarsa learns the Q-value based on the action performed by the current policy instead of the greedy policy.

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + a[R_{t+1} + \gamma Q(S_{t+1} + A_{t+1}) - Q(S_t, A_t)]$$

The action (A_{t+1}) is the action performed in the next state (S_{t+1}) under current policy (π). From the equation above (Sarsa) one can notice that two action selections are performed, which always follows the current policy. By contrast, the Q-learning equation has no constraint over the next action (A_{t+1}), as long as it maximizes the Q-value for the next state. Therefore, Sarsa is a deterministic policy algorithm (Sutton & Barto, 2018).

2.3.8 RL Special Cases

2.3.8.1 Fully Observable Environment

In this case the agent can directly observe and is aware of all the possible states in the environment (Kulkarni, 2012), thus qualifying the system as a Markov Decision Process (MDP). RL has been established to solve an MDP problem in order to come up with an optimal behavior for an agent to interact properly with an evolving environment. An MDP can be defined as a finite Markov Decision Process (a finite MDP) when the number states and actions are finite ($S < \infty, and\ A < \infty$) (Kulkarni, 2012). A finite MDP is a tuple *(S, A, T, R, γ),* with *S* denoting the finite set of environment states, *A* the finite set of actions, and T the state transition probability function from S_t to S_{t+1}, R the reward function, and γ the discount factor. A finite MDP is pivotal in understanding reinforcement learning and as a result is known to be critical to the theory of reinforcement learning (Fathi et al., 2013; Sutton & Barto, 2018).This particular case will make it possible for an agent to predict the transition probabilities *T,* which is the probability function of every potential next state S_{t+1} starting from the current state S_t

$$T[S_{t+1} = s' | S_t = s, A_t = a] = P[S_{t+1} = s' | S_t = s, A_t = a] \rightarrow [0,1]$$

On the same concept, the finite MDP has the ability to expect the value of the next reward R_{t+1} by taking into consideration the current state S_t and action A_t together with the next state $S_{t+1};$ this can be called the reward function.

$$R_{t+1}(S_t, A_t, S_{t+1}) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s']$$

2.3.8.2 Partially Observable Environment

The transition probability function and the expected reward function are crucial to the dynamic nature of a finite MDP, but optional to RL. According to Silver (2015), it is optional and not a requirement to build a model to predict the transition probability function of the next state S_{t+1} as well as the function of the expected reward. Most RL cases are based on a model-free method that assumes that the transition probability and reward functions are unknown to the agent, and the agent must interact with its environment to optimize its policy.

In this case, he Markov Decision Process (MDP) is not applicable since in many cases the agent or the learner indirectly observes the environment and does not get to see all possible states in the environment, because the environment may contain irrelevant data (Pyeatt, D., 1999). Partially Observable Markov Decision Processes (POMDPs) make it more general than the MDP to model partially observable environment (Dornheim, Link, & Gumbsch, 2018). In this situation the agent's job is to build and construct its own state representation by either remembering the complete history to have the agent state equal to the history $S_t = H_t$ or by building beliefs of the environment state. One way to do that is by using the probabilistic Bayesian approach (Pyeatt, D., 1999).

Chapter Three: Literature Review

3.1 Introduction

This chapter of the dissertation evaluates and reviews the current literature on identifying sequential sampling and reinforcement learning. This section of the dissertation will also focus on the literature reviews on researched articles pertaining to the underlying goal of this paper, which is the integration of reinforcement learning and sequential sampling methodology to help improve and aid in the successful implementation of acceptance sampling plans in an organization. This chapter also reviews the merits and demerits of industrial manufacturing, the cost of quality, the history of modern quality, sequential acceptance sampling and reinforcement learning to help give meaning to the importance of these topics to the core problem statement and objective of this dissertation. The concluding remarks on the various selected research articles for the paper will be in Chapter 4.

3.2 Documentation for The Literature Review and Keywords

In the process of gathering information for this research paper, the following online library resources, journals and electronic textbooks were consulted: University of Wisconsin-Milwaukee library database, ProQuest, Journal Storage (JSTOR), IEEE Xplore Digital Library, Informs, sinceDirect, and Compendex. The following key words will be present in this chapter: quality, lean six sigma, six sigma, lean six Sigma's CSFs, sequential sampling, acceptance sampling, and reinforcement learning.

3.3 Sequential Sampling

During World War II many manufacturing industries witnessed a major shift to the application of various sampling techniques for batches and or lots of products due to the mass production as a result of high demands for military equipment. The increase in production at various manufacturing sites and the demand for good quality products made it extremely difficult for companies to perform unit-by-unit or 100% inspection of products (Cudney, Qin, & Hamzic, 2016; Jamkhaneh & Gildeh, 2013). As a result, quality control through acceptance sampling has been used in many applications and continues to be under study. In practice, optimizing acceptance sampling techniques have widely been considered to determine the best technique for lot sentencing (Fernández, 2015).

Wu, and Liu (2014) clearly mentioned that acceptance sampling has extensively been used to determine whether to accept or reject the lot under study or inspection. The authors believe the methods of acceptance sampling plans can be one of the most well-known practical methods for quality control and quality assurance applications. Accordingly, even though there are several ways to categorize acceptance sampling plans, the authors pointed out acceptance sampling by attribute and acceptance sampling by variables as one of the major categorizations of acceptance sampling.

Much of the current studies and literature on sequential acceptance sampling have been focused on sequential sampling based on fuzzy sequential sampling. Jamkhaneh, and Gildeh (2013) aimed their work to shine new light on acceptance sampling by integrating item-by-item sequential sampling with a sequential probability ratio test for fuzzy hypotheses testing. In the process of building the authors' new model, they developed Sequential Probability Ratio Test (SPRT) for fuzzy hypotheses where the acceptable quality level (AQL) and the lot tolerance percent defective (LTPD) were considered to be the imprecise parameters. Since AQL and LTPD discussed in the article are imprecise, they tend to follow the fuzzy SPRT. However, if the authors' model has a firm AQL and LTPD, then the model is likely to follow the traditional sequential sampling plan (SSP). Afshari and Gildeh (2017) have also studied item-by-item sequential sampling based on fuzzy SPRT. However, the authors proposed a modified attribute sequential sampling plan for fuzzy hypothesis testing (FHT) with crisp observations. They then proceeded to study the impact of the ambiguity amount of defective items on the acceptance and rejection regions on their modified SSP.

Fallahnezhad, Babdi, Moeni, Sayani, and Akhoodi (2015) developed a new acceptance sampling model by using a well-known mathematical approach called dynamic programming procedure to optimize sequential sampling plans. The major objective of their study was to optimize decisions that reduce the cost associated with rejecting a batch, inspection of a batch as well as the cost of nonconforming items. In 2016, Fallahnezhad and Babadi claimed in their paper that a huge number of products are produced daily from different companies that makes it impossible for the buyers to inspect each item in the received lot. Building upon their study in 2015, Babadi tried to optimize the decision process to accept or reject or continue sampling based on the cost analysis of the lot under study by applying dynamic programming and Bayesian inference. First Bayesian inference modelling was used to predict the probability distribution of the nonconforming proportion of the lot. Later, dynamic programming was implemented to come up with the optimal decision. This dynamic programming was used when the inspection process system was imperfect and when it was perfect. The inspection process can be considered as imperfect when the producer's risk and consumer's risk (type I and type II) have an impact on the process; otherwise the inspection process is perfect.

Kumar and P. C. (2016) have also studied acceptance sampling but with a different approach. The goal of their study was to find an optimal and most efficient acceptance sampling plan that is solely based on focusing on the average life of products in a sample to make a decision to either accept or reject the lot. In order for the authors to establish their new model, a sequential sampling plan and a repetitive group sampling plan were considered. The idea behind using a sequential sampling plan was to determine when a lot can be accepted or rejected or continue sampling based on the time between successive failures (Y) and the parameters are obtained via an optimization problem for total cost reduction. On the other hand, a repetitive group sampling plan that was introduced was used and tested under four different measures. The four tested measures include the minimum of observations in a random sample, the maximum of observations in a random sample, Type I censoring, and lastly the maximum power of the test (i.e., minimizing the probability of type II error). Kumar and P. C. (2015) believe the proposed model, acceptance sampling based on the lifetime of units, will reduce the cost of inspection up to 50%.

The study goal of Fudenberg, Strack, and Strzalecki (2015) was to build a *choice process* that is based on how the probability of a more frequent choice varies with respect to the time it takes to make a decision on a task. Fudenberg et al. (2015) tested their new drift diffusion model (DDM) on a neuroscience choice experiment where the decision time in a binary choice task was explored. This model is considered as a solution to a problem of optimal sequential sampling where the agent does not have access to the outcome of its action taken in each step, and cost is involved in each sampling test. In other words, this model allows an agent to accept one alternative over another not only based on the time the agent spends to gather information, but also by the cost associated with the time spent in making a decision.

Chick, Forster, and Pertile, (2015) in their research article also agreed on the popularity of sequential sampling and how it has been implemented in various sectors such as simulation, e-commerce, and clinical trials. However, Chick et al. (2015) argued that the classic sequential acceptance sampling concept that assumes that a decision can be made right after collecting a sample without any delay might not be applicable especially in the context of clinical trials. For example, clinical trials, according to the authors, often take into account the delays between the time a decision is made during sampling and the time that the results from that sampling is finally observed. As a result, Chick et al.'s paper was focused on extending and improving sequential sampling from simulation optimization to deal with situations where delays in observing data from sampling are involved with specific emphasis on situations when the sampling variances are unknown. Chick et al. used a Bayesian model to support the case of unknown variance using a published clinical trial.

In the article by Shadlen & Shohamy (2016), the authors focused on making informed decisions based on a sequence of samples of evidence from memory. For the authors to establish their concept, they present the notion that sampling past experiences from memory aids in making value-based decisions and taking appropriate actions in the future. The authors in the article focused on linking memory to sequential sampling and experimented with their model using monkeys by studying how past rewards influenced their decisions.

Fernández (2015) focused on the design of an inspection plan for nonaccepted lots to be resubmitted for resampling inspection using two approaches, namely, the Poisson defect count and prior knowledge of the samples submitted for inspection. The author applied and studied his approach in the glass manufacturing industry. In Fernández's approach, the submitted lot is only accepted when the number of nonconforming units in a single sample is small. If not, then the lot is subjected to lot sentencing at a fixed number of times fails to relay the information. The author claimed assessing information received from previous inspections with its prior probability distribution of an unknown defect rate per unit is more beneficial and logical for the future inspection process.

Section 3.4 is dedicated to a literature review of reinforcement learning. The linkage between RL and sequential sampling will be presented in Section 4.1.

3.4 Reinforcement Learning RL

The theories and applications of reinforcement learning are much less studied compared with the other types of machine learning methodologies. However, researchers have been trying to use reinforcement learning in different sectors such as health care, economics, entertainment, and manufacturing (Gatti, 2015). In this dissertation we consider the application of RL in the field of manufacturing and particularly process or quality control. Other studies have used RL in manufacturing as well. For example, in scheduling, Ou, Chang, Arinez, and Zou (2018) pointed out that there are increasing trends of automation in today's manufacturing arena to meet the quality requirements of a product as well as satisfying increased efficiency in manufacturing processes. For instance, gantry robots, which were initially widely used in production systems for part processing, are now being used . They therefore propose an optimal gantry moving policy by using a reinforcement technique, namely Q-learning, to maximize the system outputs of the gantry-based work cell. Q-learning is known to be one of the essential techniques of RL, and it can be used to optimize the action-selection policy for any Markovian decision process. The main objective of formulating and elevating the gantry scheduling was to increase the efficiency of the work cell's real-time performance of the gantry during the production operation. The same gantry optimization approach can be used for gantry-based material handling, product transportation and quality inspection systems (Ou et al., 2018).

Shahrabi, Adibi, and Mahootchi (2017) combined reinforcement learning with a quality factor algorithm (Q-factor) to optimize the process of scheduling for dynamic job shop scheduling (DJSS). The authors implemented reinforcement learning in their model because it has the ability to optimize the process of scheduling by continually improving the policy for choosing the best parameters at each scheduling point.

Stricker, Kuhnle, Sturm, and Friess (2018) argued that there is a massive amount of data gathered and it can be useful especially in manufacturing industry, but unfortunately most of the data goes unused. Not using the available data and building a strong database can cost a company its competitive edge in its market due to lost capabilities to increase the production performance and the control process. Stricker et. al. (2018) optimized RL based on an adaptive control system for order dispatching and applied it in the semiconductor industry. The authors defined dispatching as "an optimization problem that aims to assign orders to resources and hence determine also the sequence and schedule of orders" (Stricker et al., 2018).

RL has also been applied in the area of inventory control. Katanyukul and Chong (2014) mention that inventory management can be seen as a sequential decision process and can be a decisive factor for businesses. Katanyukul and Chong's study was aimed at increasing the efficiency and flexebility of the inventory process by solving an inventory management problem using ruminative reinforcement learning (RRL). RL in its traditional form does not require domain knowledge but using RRL helps harness such knowledge of a problem's structure to improve the learning quality and speed of RL with respect to inventory management. The model proposed in this case, which is RRL, is driven by how humans ponder the consequences of their action in trying to learn how to make an optimial decision. In their study, two methods of RRL, namely ruminative state-action-reward-state-action (RSarsa) and policy weighted RSarsa (PRS), were discussed. RSarsa is portrayed to be "fast learning but leads to an inferior learning quality in the long run" (Katanyukul and Chong, 2014). PRS is portrayed to have a "suporior learning quality in the long run, but with a slower rate" (Katanyukul & K. P. Chong, 2014).

Kara and Dogan (2018) concentrated on using Q-learning and Sarsa policy to determine the optimal inventory process of product that can easily be damaged or have short lifetime. To achieve an optimal inventory process, the authors implemented their model in a manner that takes into consideration the balance between outdating a quantity of perishable products and the shortage quantity in a stochastic market.

Controlling process is another attractive area for improvement in manufacturing where RL has been implemented. Nassima, Bouziane, and Amine (2013) emphasized the importance of manufacturing control, focusing on developing a distributed dynamic control model for flexible job shop (FJS). Nassima et al.'s model was based on heuristic and reinforcement learning.

Dornheim, Link, and Gumbsch (2018) studied RL and considered it to be a potential quality tool for sequence decision making in manufacturing process. Dornheim et al. modeled a control system for nonlinear sequential processing of workpieces with discrete variables using model-free optimal control based on reinforcement learning. The proposed model was formed as a Markov decision process with a partially observable environment. A Summary of findings from the literature review will be presented in Section 4.1.

Chapter Four: Summary of Literature Review, Research Goals and Objectives

4.1 Literature Review Findings and Linkage of RL to Sequential Sampling

After an extensive background review of the history of manufacturing and quality, the results show that machine learning is part of the 4th industrial revolution and sequential acceptance sampling is part of lean six sigma, which is the latest quality technique in modern manufacturing (arguably as well as in the service industry).

From the presented literature, manufacturing industries are currently more focused on finding ways to improve product quality and variety (process agility) and produce more products within a short lead time. As a result, quality control and optimizing acceptance sampling techniques, particularly sequential sampling, have been considered in many studies. Noticeably, the reviewed articles showed that sequential sampling plans are the focus of many studies, most of which try to improve the sampling policies by reducing the sample size and eventually the production cost. The literature review shows several proposed approaches to optimizing sequential sampling plans based on integrating different techniques, but none use the reinforcement learning approach. As a summary, some of the techniques used in the literatures for sequential sampling optimization include:

1. Fuzzy sequential sampling
2. Dynamic programming
3. Average life of products in a sample
4. The Bayesian methodology
5. Poisson defect counts
6. Prior knowledge of the samples submitted for inspection (Sampling based on memory).

We therefore believe that with this dissertation we present a novel RL-based approach to sequential sampling optimization.

After reviewing the RL literature, it is certain RL has had a major role in revolutionizing the manufacturing industry with applications such as, but not limited to, sequence decision making optimization, quality control, supply chain, scheduling, inventory, and production optimization, but not many studies have been conducted to study the integration of RL and acceptance sampling.

One of the major attributes or components of reinforcement learning is a reward system that guides an agent to optimize its decision making towards its next action. In previous studies, system reward and recognition have been considered as a major key for RL but there has not been a study that directly links RL to sequential sampling. This dissertation will attempt to improve sequential acceptance sampling with RL and investigate its impact on the process of inspection. In another words, this dissertation aims to consider an improved sequential acceptance sampling plan with RL as an optimal inspection plan.

Dornheim et al. (2018) claimed that in every manufacturing process, the process quantities measured do not precisely define the state of the process with regards to an optimization problem. This claim does not support the idea of LSS where the processes have to be clear and fully observable. For example, in the process of acceptance sampling that is the focus of this dissertation, managers, operators, and inspectors have prior knowledge about quality parameters of the produced products and whether the products (lots) were accepted or rejected. That is to say, the proposed integration between RL and sequential sampling will consider the process of inspection as a fully observable environment for inspectors and follows MDP.

4.2 Research Goal and Objectives

The overarching purpose of this dissertation is to improve the process of sequential acceptance sampling using the RL methodology and to examine the impact of the proposed model on the LSS strategy for a high-volume manufacturing system. The following research objectives will be accomplished to achieve this goal:

Objective 1: Build a sequential acceptance sampling model using the RL methodology for a partially observable environment. We hypothesize that by incorporating the RL methodology, the sequential sampling procedure will be improved in comparison with the traditional sequential sampling method.

Objective 2: Carry out a sensitivity analysis.

Objective 3: Compare the results of the RL-based sequential sampling with other sampling techniques, such as the fixed and sequential sampling techniques as well as the MIL-Standard policies.

Chapter Five: The Methodology

5.1 Introduction

The concepts of sequential acceptance sampling and reinforcement learning are introduced and discussed in Chapter 4. Sequential sampling is a quality control technique in which time and cost can impact the process of product quality inspection in several ways. First, defective items that are detected may be scrapped or rerouted for rework, the cost of which dwindles in comparison with the cost of defective products being released into the market. Second, an increase in the number of items to be tested (sampling size) increases the cost of testing and especially when such tests are destructive. Third, the sampling time also adds onto the cost of sampling. That said, sequential test policies enable the sample size and inspection interval to be dynamic, in response to the current known system quality state. It is for this reason that we propose using an RL-based sequential sampling technique.

RL is a self-learning technique that ensures optimal agent action in the future. We hypothesize that modeling sequential acceptance sampling using the RL methodology will result in reduced cost and time in the process of implementing LSS to ensure that quality products are been produced.

The proposed model is a comprehensive RL-based sequential sampling optimization technique for a high-mix, low variety manufacturing facility. The proposed model focuses on the sample size, the sampling frequency, and the sampling interval, all of which are parameters considered by a quality inspector when making a decision on whether to accept or reject products in an assembly line. In this case, the inspector bases his or her decision on the observations and rewards received from the previous inspection.

Since the process of the proposed model has randomness in its operation, it is a stochastic problem. The inspector in the proposed model has no previous knowledge about what product is good or not. Moreover, this model assumes the inspectors can fully observe the products under inspection and determine if they are acceptable or should be rejected based on their defects. The overarching purpose of this dissertation is to improve the process of sequential acceptance sampling by using RL and to examine the impact of the proposed model on the sampling process. The results from this novel model will determine whether linking RL with sequential sampling can be considered as an optimal solution to sequential sampling. The second part of this chapter begins with explaining the most related criteria to the proposed model. The third part illustrates the steps of the proposed model. The fourth part simulates the presented model using Python programing. Lastly, the results and conclusions are presented.

5.2 Models Description

The proposed model is based on discrete manufacturing processes that produce hundreds or thousands of products a day. In such cases, 100% quality would be nearly impossible to achieve. To ensure that quality products are being produced at low costs while achieving customer products' requirements and satisfaction, the majority of these manufacturing companies adopt different sampling techniques to aid and streamline the inspection process for products coming off the production line. These sampling techniques or inspection plans often involve selecting samples off the production flow and spending some time (inspection time) to perform quality inspections for defects.

This dissertation introduces a model that addresses some inefficiencies with the sampling techniques. To reiterate, inspectors or decision makers are the model agents, who are trained and have complete knowledge of how to inspect each item, as well as full control to decide when to continue sampling, stop sampling, determine when to take the next sample, and the number of samples. The sampling process is the agent's environment and it is stochastic and discrete.

The current number of samples and the sampling interval depend on the number of nonconformities observed from the previous inspection. The sampling plan used in this scenario is a sequential sampling plan and it is considered to follow a Partially Observable Markov Decision Process (POMDP), where each state in the process of sampling is partially observable by the inspector at each timestep.

5.3 Model Notations

The problem here is to find the optimum sampling policy regarding the sampling operation that exists in the production line by using reinforcement learning and sequential sampling as a new acceptance sampling plan. The production process considered in this dissertation is a continuous process and requires selecting a small number of samples due to the cost and inspection time. Overall, the quality control process of the proposed model depends on the sequential sampling hypothesis, which is:

$$H_o: p \leq p_1$$

$$H_a: p \geq p_2$$

Where p is the quality parameter of a lot under inspection, p_1 is the acceptance point, and p_2 is the rejection point. As stated earlier, sequential sampling is controlled by the predetermined value of the producer's risk (α), the consumer's risk (β), and the acceptance and rejection points p_1 and p_2, respectively, are estimated as follows:

$$p_1 \approx \log \frac{\beta}{1-\alpha}$$

$$p_2 \approx \log \frac{1-\beta}{\alpha}$$

The log-likelihood function in this matter follows a binomial distribution and uses the number of nonconformities p found in sample n to test its hypotheses, where r is the number of non-confirming products, and n is the sample size.

$$log\,\Delta\,(r,n) = log\frac{f_{p_2}(r,n)}{f_{p_1}(r,n)}$$

The quality decision in the proposed model is considered as the agent action and is controlled by the cumulative log-likelihood ratio S_i.

$$S_i \ = \ S_{i-1} + log\Delta$$

If S_i is less than or equal to the acceptance constant p_1, the lot is accepted. If the S_i is greater than or equal to the constant rejection p_2, the lot is rejected. However, if S_i is in between p_1 and p_2, then another sample must be drawn (Otieno & Nanduri, 2012).

This work aims to optimize the sample size and the sampling frequency from using sequential sampling and by using the reinforcement learning-based MDP solution.

Below are the criteria of the agent process:

1- S is a set of environment states $S_t \in S$, and it is defined by whatever information is available to the agent during each inspection at each discrete time step t. In the model, the available observed data to the agent during each state are characterized by the proportion of defectives (p_t). Thus, the state space is presented as $S_t = \{p_1, p_2, \ldots\ldots, p_t,\}$.

2- A is a set of possible actions $A_t \in A$ at each state S_t, where the available actions for the agent at each time can be defined as $A_t = \{(g_1, k_1, w_1), (g_2, k_2, w_2), \ldots\ldots, (g_t, k_t, w_t)\}$ where g_t is the action of taking a sample, k_1 is the action of stopping the process, and w_t is the action of waiting for later sampling. The agent policy is a stochastic policy denoted as $\pi_t (a|s)$ that is mapping the probability of each available action given the current state S_t. At each state the agent follows a policy to select an action from a set of actions $A_{t+1} \in (g_t, k_1, w_t)$ that maximize the expected return modeled as $\pi^* = argmax_A\, R\ (S_t, A_t)$. The optimal policy is based on the proportion of defectives p_t observed from each sample S_t (environment's state) and the reward R_t associated with it.

3- R is the reward distribution function, and it can be defined as the immediate reward the agent will receive after each inspection at t time. R_t is the reward received at each state S_t after executing action A_t. The reward R_t for this study is a policy based on the proportion of defectives p_t observed from each sample S_t and the sampling interval (w_t). Roughly speaking, the total rewards for the proposed model are saving cost and time as well as we assume that all the immediate rewards are positive.

4- To minimize the risk associated with each available action such as deciding to continue sampling in a fixed time that will cost money and time, especially in the process where LSS is implemented and the level of the quality process is high. Q-learning is a value-based learning method used in reinforcement learning that helps the agent to find the optimal action-selection. The Q- learning algorithm

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + a[R(S_t, A_t,) + \gamma\ max_a\ Q(S_{t+1} + A) - Q(S_t, A_t)]$$

5- T (S_t, A_t, S_{t+1}) is the transition probability function in the transition between one state S_t to another state S_{t+1} after the agent executing action A_t. The transition probability function in the proposed model is under the influence of the predetermined parameters of sequential sampling viz, the producer's risk (α), the consumer's risk (β), and the acceptance of a rejection point p_1 and p_2 respectively. For example, if the observed data (number of defects) is greater than or equal to the rejection quality level, then the probability of stopping the process for further investigation will be 1.

6- w (S_t, A_t, S_{t+1}) is the transition time between one state S_t to another state S_{t+1} after the agent executing action A_t. The transition time between states represents the time the inspector has to wait for taking another sample.

7- Gamma (γ) is the discount factor that helps the decision makers to determine the expected value of the future reward received from the future sampling. The value of gamma is between 0 and 1 ($\gamma \in [0, 1]$); the higher the value of γ, the less the agent is discounting.

5.4 Model Formulation

In this section, we will review the general steps and the algorithms after combining reinforcement learning with sequential sampling:

1- We set Q-factors for all state-action pair and value function V to 0. Thus, there are 3 actions for each state where $A_t \in (g_t, k_t, w_t)$, and state $S_t \in S$. Thus, for each $Q\ (A_t, S_t) \leftarrow 0,\ Vt\ (A_t, S_t) \leftarrow 0$.

2- The agent takes an action $A_t \in A$ at time t from the initialized state S_t with a probability of $\frac{1}{A(S_t)}$.

3- The optimal action-value function A_t is calculated

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + a[R(S_t, A_{t},) + \gamma\ max_a\ Q(S_{t+1} + a) - Q(S_t, A_t)]$$

4- Agent observes the number of defects and reward R_t from state S_t after executing action A_t, let $R\ (S_t, A_t, S_{t+1})$ *to reduce the sample size and wait time.* For instance, the observed reward at each state followed a model

If $p_t < p_{t-1}$ *and* $w_t > w_{t-1} \rightarrow R_t = 1$

If $p_t = p_{t-1}$ *and* $w_t = w_{t-1} \rightarrow R_t = 0$

If $p_t > p_{t-1}$ *and* $w_t \leq w_{t-1} \rightarrow R_t = -1$

5- Agent updates the total reward: *total reward* $\leftarrow w\ (S_t, A_t, S_{t+1})$

6- Agent updates the function $Q\ (s, a)$

   a. Exploration if it is larger than lambda (λ)

   b. Exploitation if it is less than lambda (λ)

   c. Continue until done

7- Agent transition time between states w (S_t, A_t, S_{t+1}) can be based on the expected probability of defects that set by Bayesian probability or based on the predefined defects limitations.

$$P(Def|S_t) = \frac{P(S_t|Def)\ P(Def)}{P(S_t)}$$

For example,

If $1 \geq P \geq 0.90 \rightarrow w_{t+1} > w_t$

If $0.89 \geq P \geq 0.80 \rightarrow w_{t+1} = w_t$

If $0.87 \geq P \geq 0.60 \rightarrow w_{t+1} < w_t$

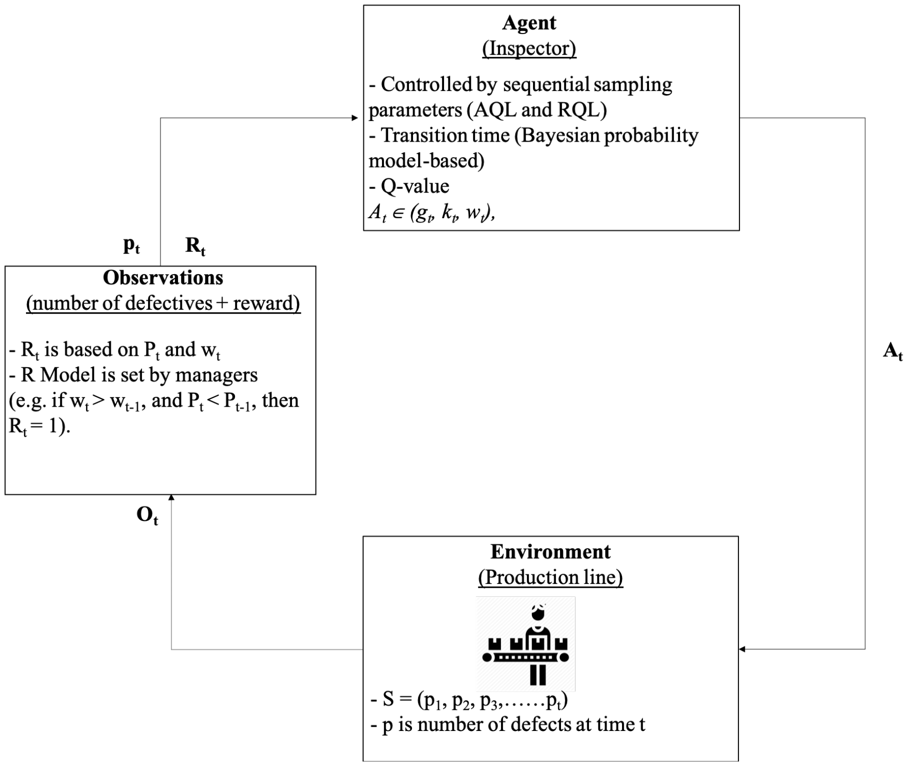


Figure 17: Sequential Acceptance Model Based on RL

Chapter 6: Sequential Sampling RL based Simulation

6.1 Introduction

An efficient production system requires models that are capable of reducing the number of samples during an inspection. While this paper has described the impact of acceptance sampling plans in industries, it is necessary to have a model that optimizes the process of sampling. It is also fundamental that the proposed model to be simulated on a real-world inspection planning problem before it is implemented in practice. This chapter presents a simulation technique based on integrating RL with sequential sampling that helps to minimize the total number of samples during the inspection process. The goal of using simulation is to give a representation of a real process over time by imitating the real production system. Therefore, the performance of the RL-based sequential sampling technique, sequential sampling, and MIL-STD-1916 is evaluated by simulation using Python programming. This software uses a collection of built-in functions and methods (libraries) that meets the users' needs to perform actions without the need for writing new code such as statistical analysis, optimization, and plotting functions.

The simulation model in this dissertation describes the inspection processes in a production line and estimates its performance for each given factor (Q, S). That is to say, the (Q, S) factor is considered as the input of the new model, and the related sample size and number of nonconformities are the output of the model. Reducing the sample size during the inspection is the primary goal of this dissertation. The RL-based sequential sampling technique will achieve that goal, and it will be compared with the MIL-STD-1916 and sequential sampling plans that were described in Chapter 4.

In order to achieve a specific production goal such as reducing the time of inspection or reducing the sample size in the implementation of any acceptance sampling plans for manufacturing industries, decision-makers have to consider and evaluate the current production system, measure the performance of the current system, and standardize the operating system procedures. For those reasons, we consider the production system in the simulation of RL-based sequential sampling, sequential sampling, and MIL-STD 196 as a stochastic environment. The dataset that is used for the simulation was created using Python; we assume there are 5,000,000 products in a production line, and they are produced following a discrete event. The number of nonconformities is assumed to be 8% for the entire batch and follows the Poisson distribution.

This section of the dissertation provides the criteria for evaluating the performance of each model. Moreover, this part will cover in detail the simulation of the RL-based sequential acceptance sampling technique in the field under study. The simulation of sequential sampling, MIL-STD-196, and MIL-STD-196 will be presented in Part Three. Lastly, the results of the simulated inspection process problem will be described.

6.2 Simulation Model (RL-Based Sequential Sampling)

Python programming was used to build and train the RL-based sequential sampling model. The Python libraries that were used include NumPy, Matplotlib, CSV, Collections, and Ransom. The model focuses on the activities of sampling during production. The assumptions made to build this model are as follows:

1. The model assumes there are continuous manufacturing processes that produce a specific number of 5,000,000 products. The beta distribution is known to be a prior distribution and finite support. As a result, beta distribution is applied to simulate the

behavior of generating random variables from observing the number of a random process that emits a series of defective or not defective items over time.

2. The model assumes that there are discrete manufacturing processes that produce a specific number of products (N=5,000,000).
3. The process is assumed homogeneous, i.e., the distribution of the non-conformance has an expected value of 0.08 for all simulated data.
4. The batch size of the reinforcement learning model is assumed to be between 400 and 15000, depending on what the algorithm learns.
5. The model assumes that the acceptable quality level of nonconformities (p_1) is 0.01, and the rejection quality level of nonconformities (p_1) is 0.06.
6. For simulation purposes, a parameter lambda (λ) is used to speed up the algorithm and to configure the Poisson distribution that controls the environment's batch sizes. In this case, the model assumes that lambda equals 0.1, the beta value (β) that is the consumer's risk equals to .10 and the producer's risk (α) is assumed to be 0.05.
7. Each inspection plan works for the entire production processes.

8- The model assumes that there are five states for the agent's space St = {S_1, S_2, S_3, S_4, S_5}.

8. During the inspection process, this model assumes there are five different states that the agent can be in, based on the number of nonconforming samples.
9. The model assumes that the agent's actions are dependent on the state in which the algorithm is.
10. All quality inspection parameters are assumed to be predefined and controlled by the sequential acceptance sampling plan.
11. The reward function for this model is assumed to be a function of (the number of samples, the accepted weight, the number of accepted bad samples, the rejected

weight, the number of rejected bad samples, and the states). This reward function can be logically acceptable for the purpose of this work, since it considers different types of samples' rates for comparison.

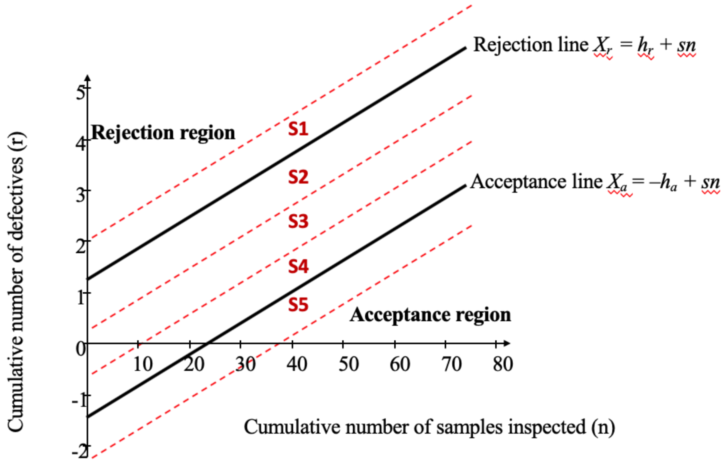


Figure 18: RL-based Sequential Sampling Model's States

6.3 Simulation Performance Measures

The main goal of the dissertation is to reduce the sample size during the inspection process. So, it is critical to find out the different rates of sampling. Therefore, we evaluate and compare the performance of each plan based on five criteria that will give us the information not only on how many samples are inspected but also the information on how many bad samples are accepted, and how many bad samples are actually rejected. The performance of each acceptance sampling plan can be measured in many different ways, as discussed in section 2.2.6.5.1. For this work the different rates of inspection will be considered and calculated as follows:

$$Sample\ rate = \frac{\#\ of\ total\ samples\ inspected}{N\ batches * batch\ size}$$

$$Bad\ accepted\ rate = \frac{\#\ of\ bad\ samples\ in\ the\ accepted\ batches}{\#\ of\ accepted\ part}$$

$$Accepted\ weight = \frac{\#\ of\ parts\ \ accepted}{N\ batches * batch\ size}$$

$$Bad\ rejected\ rate = \frac{\#\ of\ bad\ samples\ in\ the\ rejected\ batches}{\#\ of\ rejected\ parts}$$

$$Rejeceted\ weight\ = \frac{\#\ of\ rejected\ parts}{N\ batches * batch\ size}$$

6.4 Model constraints

The purpose of the algorithm is to determine a more efficient method of sampling in order to use fewer samples to determine if product made will meet the required manufacturing quality specification. Having constraints for the problem will make the algorithm more efficient because it decreases the size of the search space. The main constraints that are used in the process of designing the simulation model include:

1. Hidden batch size min/max (400 samples/15,000 samples): is one of the parameters that the agent is trying to learn. Basically, the performance of the model is determined by the hidden batch size and the cut-off point for the model. Since the parameters are learned, it enables the model to have a higher performance than simply taking the batch size value from a table. The hidden batch size considered in this model is between 400 samples and 15,000 samples.
2. Sample size min/max (11 samples/1111samples): the minimum sample size that the agent can inspect from each batch is 11samples, and the maximum sample size is 1111.

3. Wait time min/max (11 samples/1111 samples): the minimum wait time that the agent can wait between each inspection

States' thresholds (S1, S2, S3, S4, S): it is essential to constrain the search space (number of nonconformities in our model) in order to make it so that the algorithm will converge faster and better. The states' thresholds are designed by first determining the rejected point and accepted point from the sequential sampling plan. Table 11 presents the number of nonconforming samples for each state threshold constraint.

Table 11: States' threshold constraints

State	Minimum	Maximum
State 1 threshold (Rejected line in SS)	0.80	0.91
State 2 threshold	0.85	0.93
State 3 threshold	0.88	0.96
State 4 threshold	0.90	0.97
State 5 threshold (Accepted level in SS)	0.92	0.98

The most general case of the algorithm would not have any constraints for determining the required thresholds in order to meet that purpose. The following cases are used to address the algorithmic convergence challenge:

1. Lack of constraints in determining the thresholds will lead to slow convergence.
2. With exact constraints, the algorithm will converge on the first iterations. In this case, there is no point in having the thresholds as a tunable parameter.

That being said, since we do not know what the thresholds should be ahead of time (this is supposed to be determined by the characteristics of the distribution of the quality of the product), the constraints can be made to be as broad as possible. Thus, in order to make them as broad as possible while also having excellent convergence characteristics in some cases, there can be an overlap between multiple state's constraints. Moreover, each state transition shows in Figure 19 is determined by what the current state is. So, the agent's transition is dependent on which state the algorithm is in and what the algorithm has learned should be the state transitions.

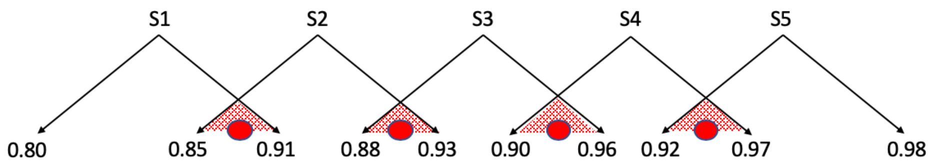


Figure 19: Overlap Between Multiple State's Constraints

6.5 Simulation Procedure

Once the constraints are defined, see part 6.4, all that remains is the setup and initialization for the algorithms following the same steps that were explained in Section 5.4. Once the algorithms are initialized, the steps can be repeated until either a fixed number of iterations is reached, or a stopping criterion is achieved. That stopping criterion is easy to define if the reward function is well understood. When the reward function is not well understood or not designed well, then there may be two options: (1) run the algorithms a set number of iterations (algorithm loop) or (2) determine a better reward function (that can be easier said than done because each organization has its own quality parameters and ways for evaluation).

After the algorithms are initialized, create the Q-table and set the Q-factors for all state to 0. We initialize the first state to state 3, and the loop of the model starts as:

1. Evaluate the current state (data environment, state, and current state). The purpose of this step to determine what the characteristics of the environment are while evaluating.

a. Get the number of the next sample that the algorithm will evaluate for the next iteration of the loop.

b. Get sample size number of samples from the data environment (only accepted samples)

c. Calculate all the values from the wait time for reward evaluation (number of samples not sampled).

d. Calculate the accepted weight from the samples:

$$Accepted\ weight = \frac{\#\ of\ parts\ accepted}{N\ batches * batch\ size}$$

e. Compare with the current best state:

If the current accepted weight is larger than the values of state 5 $\rightarrow$ the state return is state 4.

Else if: the current accepted weight is larger than the value of state 4 $\rightarrow$ the state return is state 3.

Else if: the current accepted weight is larger than the value of state 3 $\rightarrow$ the state return is state 2.

Else if: the current accepted weight is larger than the value of state 42$\rightarrow$ the state return is state 1.

Else if the state return is not equal to the current stat$\rightarrow$ break

else: the state return will remain the same.

2. Calculate the current state reward (a function of (number of samples, accepted weight, accepted bad samples, rejected weight, rejected bad samples, states). In order to reduce the number of sample size and motivate an agent to inspect fewer samples, each sample will have a value of -1 to the total reward of the current state. The reward equation is

-1 per sample + number of good samples per correctly accepted batch – size batch if rejected – (10 x number of bad batches incorrectly accepted) + 100:

If the current state is state 1 and the bad accepted rate > 0.1

If the current state is state 2 and the bad accepted rate < 0.1

If the current state is state 3 and the bad accepted rate < 0.0875 $\rightarrow$ new reward +=100

For state 4

If the current state is state 4 and the bad accepted rate < 0.075

for state 5

If the current state is state 5 and the bad accepted rate < 0.06

If any state and the bad accepted rate < 0.05

3. Compare the current reward with the best reward; if the reward is higher than the best reward, then the best state is set to the current Q value and the best reward is set to the current reward.
4. Current reward is set to 0
5. A random number is generated and compared with lambda
6. If the number is greater than lambda, then the algorithm will be set into exploit mode, meaning the constraints for generating the next state are set to be close to the current best state (exploit).
7. If the random number is less than lambda, then the global constraints are used for setting the next state (explore).
8. Once the new state is generated, we repeat the loop, or we stop if the policy meets objectives.

A summary of the simulation process is shown in Figure 20

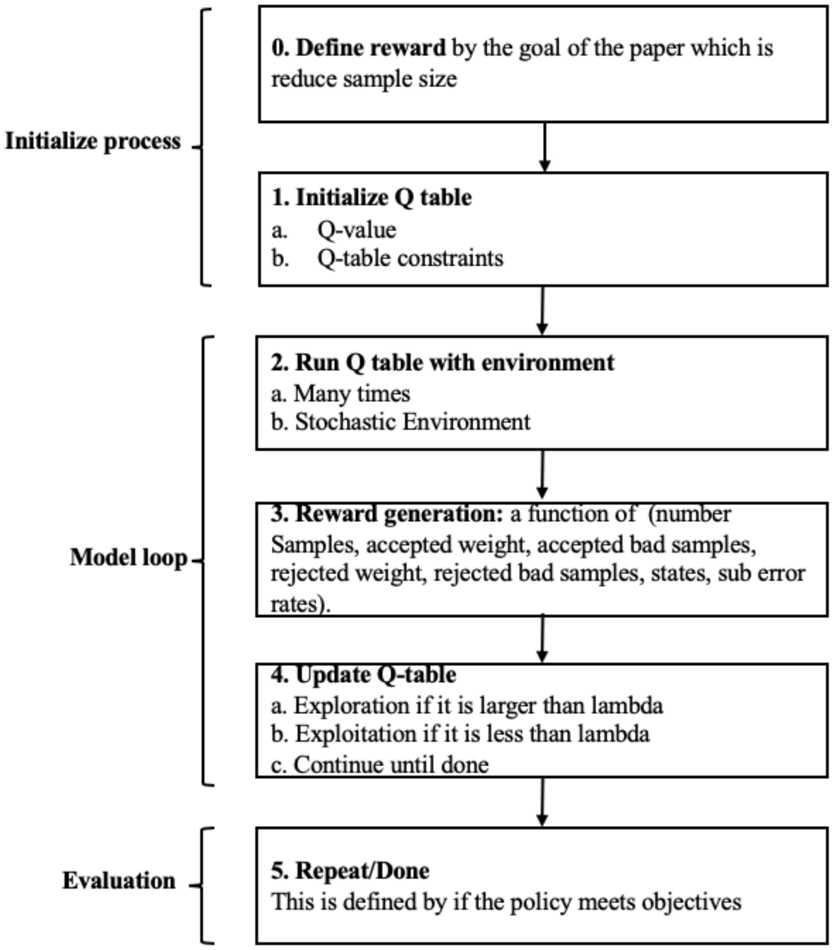


Figure 20: A Summary of RL-based Sequential Sampling Simulation

6.6 RL-Based Sequential Sampling Results

After three different trials of the simulation to estimate the performance of the proposed model in Part 6.5, Table 12 shows the average of the best three results from 12 hours of running the algorithm. From the same table, the sample rate of run 3 was 16%, which is higher than the other two. From the simulation results, there is no statistical difference in the bad accepted rate between the three simulations. However, the rejected weight for run 2 is 14%, and it is higher than the other cases. Table 12 also shows that the accepted weight of the three runs appeared to have a 3 percent difference between the lowest and the highest. Additionally, there is clear evidence that there is a direct correlation between the sample rate and the accepted weight, where, when the sample rate increase, the accepted weight increased as well. Figure 22 represents the trend of trial 2 results. The trends of trials 1 and 3 are very similar to trial 2 and hence are not included. We note that the average values in Table 12 are for comparison between the RL model and the other sampling plans (Chapter 7).

Table 12: Results of the 6 hours simulation RL-based sequential simulation in %

	Sample rate	Bad accepted rate	Accepted weight	Bad rejected rate	Rejected weight
Run 1	7	10	97	12	3
Run 2	4	10	94	14	6
Run 3	16	10	96	12	4
Ave.	9	10	96	13	4

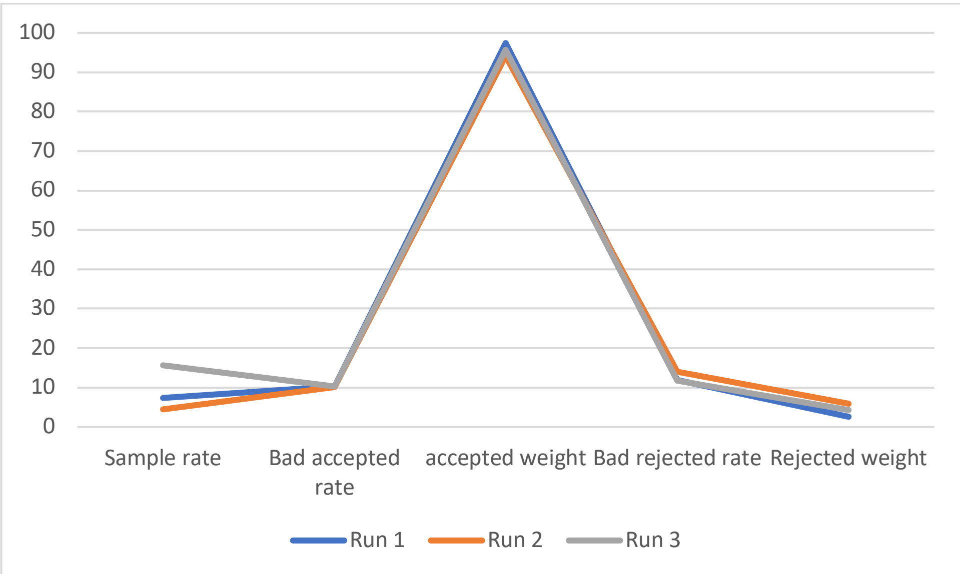


Figure 21: Results of Similation1, 2, and 3 for RL-based Sequential Sampling

Figure 22 shows an example of how the algorithm has randomness in its operation. This randomness is due to the agent policy, where it has to explore the environment to accumulate more rewards. For the agent to obtain more rewards, it has to sample fewer and detect more defectives at the same time. There presents an opportunity for the model to promote the agents' exploitation rather than exploration, which would potentially increase the number of the bad accepted rate with the minimal sampling rate possible. This idea will be discussed in the future research section.

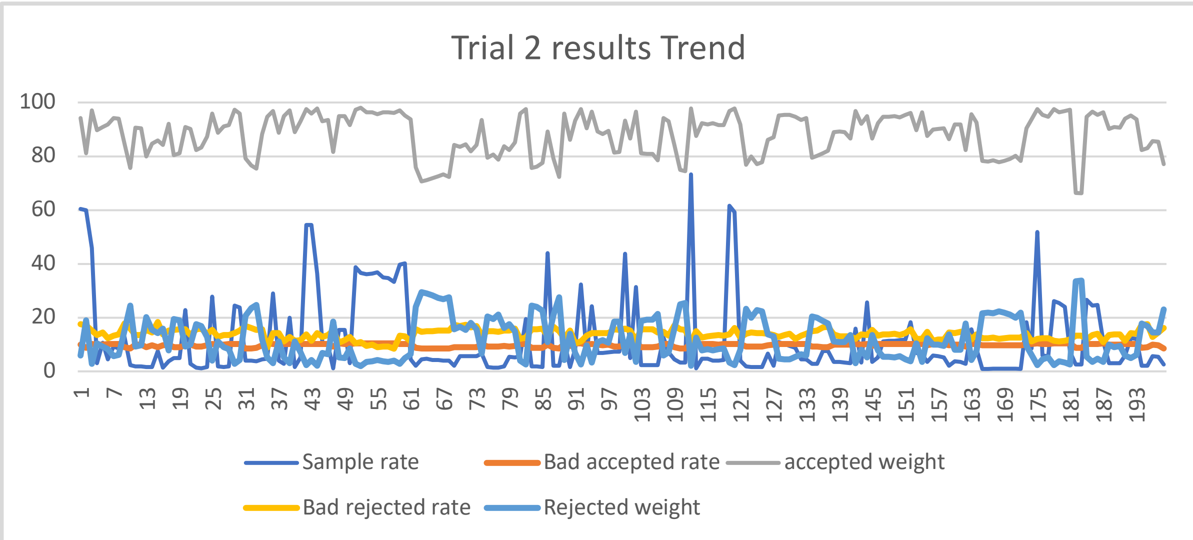


Figure 22: RL-based Simulation Behavior

Similarly, Figure 23 shows that the sample rate is always increasing at the beginning of each simulation, and later decreases. This is because the agent policy promotes the exploration of the environment more than taking the same action in each state. Similarly, the algorithm is detecting more defectives at the end of the simulation than at the beginning, because later in the algorithm, the agent has already built knowledge on how the environment works.

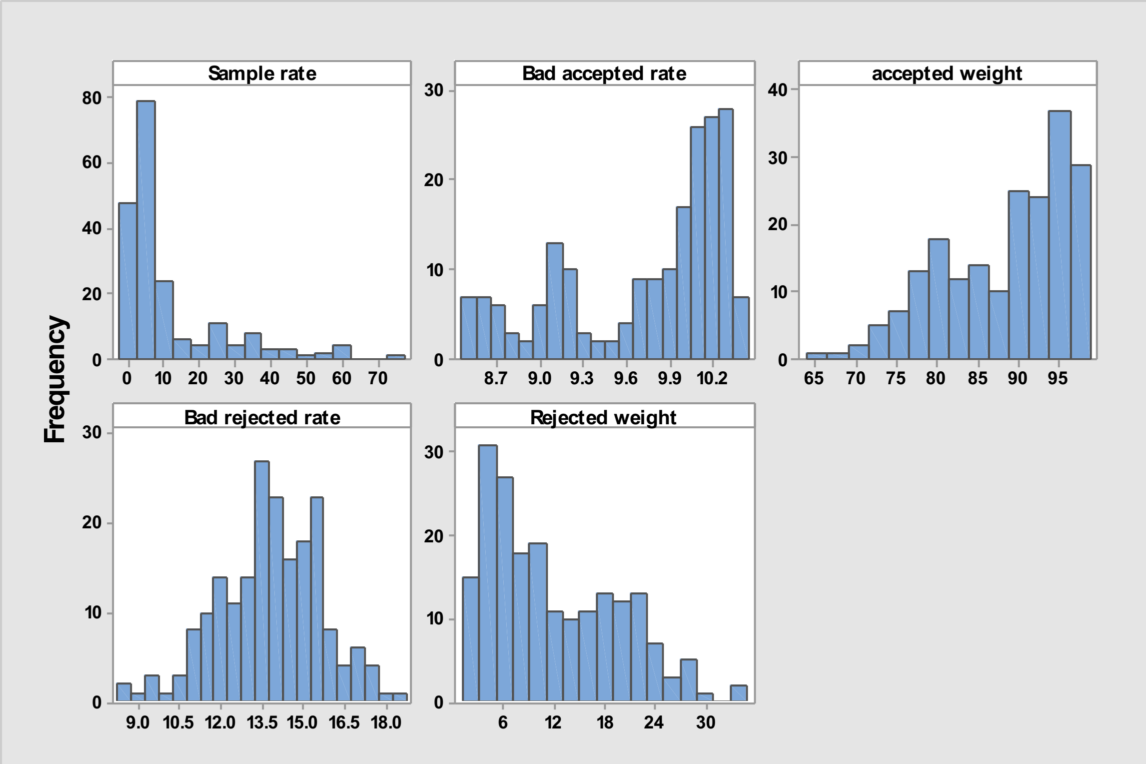


Figure 23: Histogram of RL-based Sequential Sampling

In Chapter 7, we consider a set of perform in a sensitivity analysis of the RL-based model by choosing to make perturbations to the model design parameters, to produce seven sensitivity analysis scenarios. In addition, we compare the results of Trial 2 of the RL model with the other acceptance sampling models, i.e., the existing sequential model, MIL-STD-1916 (32), MIL-STD-105E (Single sampling), MIL-STD-105E (Double sampling), and MIL-STD-105E (Multiple sampling).

Chapter 7: Numerical results and Model Comparisons

7.1 Introduction

Chapter 7 aims to perform a sensitivity analysis to identify and test the behavior of the RL-based sequential sampling model under different parameter value conditions. This chapter describes some empirical evidence of the advantages of using the RL-based sequential sampling over the other acceptance sampling plans. In Chapter Five, the formulations for RL-based sequential sampling model were provided. In Chapter 6, conceptual design parameters such as the beta value, lambda value, and state's constraints were provided in the design of simulating RL-based sequential sampling. The simulation then interacts in a manufacturing process specifically in the process of inspection. This section shows that RL-based sequential sampling can adjust to a different level of parameters. Later in this chapter, comparisons are presented between the simulation model described in Part 6.2 to sequential sampling, ML-STD-1916, and MIL-STD-105E. For a visual representation of the results of the comparison, the reader is referred to Table 25.The data and their graphical display gathered for each simulation are too large; thus, they are not included in this chapter, but they are included in the appendix.

Moreover, to perform the sensitivity analysis, one or two factors occur in each case, while the other factors remain the same. The factors considered in these sensitivity analyses are the states' thresholds, beta value, lambda value, sample size, and batch size. Generally, when a change is made to one model parameter, all the rest are kept constant. The following is a discussion into each model parameter that is used in the sensitivity analyses:

1- State thresholds constraints.

In the RL-based sequential sampling model, *overlap* parameters are used to constrain the search space (states' thresholds), to enable the algorithm to converge faster (Figure

19). For testing the performance of the proposed model, fixed thresholds parameters are implemented as shown in Figure 24.

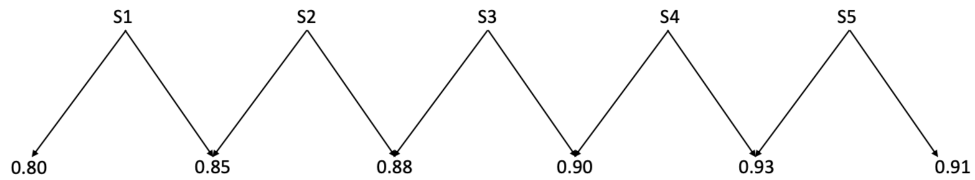


Figure 24: States' Exact Thresholds Constraints

2. Beta value

One way to study the performance of the proposed model is to try to change the parameters in the sequential sampling plan, such as the rejectable quality level and the beta value (consumer's risk) that represents the probability of sending a lot with several defectives that exceed the rejectable quality level to the consumer.

3. Lambda value

Lambda (λ) is a value between (0, 1), and it is part of the algorithm to speed up the process of learning. The lambda value helps the algorithm to accelerate and learn faster in each state instead of going back to the system that had been created. Lambda is also used to configure the Poisson distribution that controls the environment's batch sizes. Three values of lambda are considered 0.9, 0.05, and 0.2.

4. Sample size and batch size

The sample size and batch size can affect the results of the simulations. As a result, the sample size is considered to range between 8 to 50, which represents the smallest sample size and the largest sample size in the MIL-STD-105E for a batch with 500 parts.

5. Reward

The agent performance depends on how the reward function works in each state. The same reward function in 6.5 is implemented but with different conditions in receiving the rewards. The reward function is defined as:

-1 per sample + number of good samples per correctly accepted batch – size batch if rejected – (10 x number of bad batches incorrectly accepted) + 100:

If the current state is state 1 and the bad accepted rate > 0.2

If the current state is state 2 and the bad accepted rate < 0.2

If the current state is state 3 and the bad accepted rate < 0.1

If the current state is state 4 and the bad accepted rate < 0.0785

If the current state is state 5 and the bad accepted rate < 0.05

If any state and the bad accepted rate < 0.05

Table 13 shows a summary of the factors that are used to perform five different sensitivity analyses, followed with a description for each factor. The results of each model and the comparisons between the models are discussed later in this chapter.

Table 13: Description of Sensitivity Simulations

Sensitivity analysis	Factor 1	Factor 2
1	Fixed State Thresholds constraints State1: 0.80 – 0.85 State 2: 0.85 – 0.88 State 3: 0.88 – 0.90 State 4: 0.90 – 0.93 State 5: 0.93 – 0.98	
2	Sample size: 8 to 50	Batch size = 500
3	Beta = 5	
4	Lambda = 0.9	
5	Beta= 5	Lambda = 0.05
6	Reward	

7.2 Data Analysis and Results

Similar to the main model, each sensitivity analysis scenario simulation is repeated three times, and each run takes an average of 12 hours. The average of the best three results in Part 6.5 will be compared with the model under different parameters and the acceptance sampling plans.

7.2.1 RL-Based Sequential Sampling Under different parameters

The first part of this analysis is to find out how the RL-based sequential sampling parameters can be critical to the results.

Table 14: Sample Rate-parameters Comparisons

Test	Evaluation criteria	Sample rate
RL	RL-based Seq. Sampling	9
1	State Thresholds constraints	12
2	Sample size: 8 to 50, Batch size = 500	22
3	Beta = 5	21
4	Lambda = 0.9	10
5	Beta= 5, Lambda = 0.05,	49
6	Reward, Lambda = 0.2	15

It is apparent from Table 14 in that the sample rate of four tests has increased by more than 50% comparing with the result from the first line. Noticeably, the highest sample rate is realized when the beta value is 5 and lambda = 0.05. That is because when lambda is too low, the model will learn excessively about the characteristics of the training data, and that enables the model to generalize to new data and actions. Where lambda is too high such as in test 4 where lambda is 0.9, the model reduces its speed and the agent does not learn enough about the observed data to make a better decision, thus resulting in fewer inspected samples.

Table 15: Bad Accepted Rate-parameters Comparisons

Test	Evaluation criteria	Bad accepted rate
RL	RL-based Seq. Sampling	10
1	State Thresholds constraints	10
2	Sample size: 8 to 50, Batch size = 500	10
3	Beta = 5	19
4	Lambda = 0.9	8
5	Beta= 5, Lambda = 0.05,	16
6	Reward, Lambda = 0.2	8

Table 15 indicates that there is a significant difference in the bad accepted rate after changing the beta values. Unlike the results obtained from the first two simulations where the bad acceptance rate was similar to the original model, a change of beta to 5 increases the bad accepted rate to 16.

Table 16: Accepted Weight-parameters Comparisons

Test	Evaluation criteria	Accepted weight
RL	RL-based Seq. Sampling	96
1	State Thresholds constraints	88
2	Sample size: 8 to 50, Batch size = 500	88
3	Beta = 5	88
4	Lambda = 0.9	88
5	Beta= 5, Lambda = 0.05,	88
6	Reward, Lambda = 0.2	93

Table 16 shows that the changes in the state thresholds parameters, sample size (min 8, max 500), beta value, and lambda value have the same impact on accepted weight at 88% compared with the original model whose value is 96%. However, Table 16 shows that there is no significant change in the accepted weight when lambda is increased to 0.2, and the new reward system is changed.

Table 17: Bad Rejected Rate-parameters Comparisons

Test	Evaluation criteria	Bad rejected rate
RL	RL-based Seq. Sampling	13
1	State Thresholds constraints	15
2	Sample size: 8 to 50, Batch size = 500	15
3	Beta = 5	25
4	Lambda = 0.9	14
5	Beta= 5, Lambda = 0.05,	22
6	Reward, Lambda = 0.2	13

Table 17 shows positive rates of change in the bad rejected rate for all the tests since defectives are detected before the lots are shipped. It also shows the highest bad rejected rate when the beta value is set to 5. Even though the beta value is changed as well as the reward in test 6, the bad rejected rate remains the same at 13% when compared to the base model.

Table 18: Rejected Weight-parameters Comparisons

Test	Evaluation criteria	Rejected weight
RL	RL-based Seq. Sampling	4
1	State Thresholds constraints	12
2	Sample size: 8 to 50, Batch size = 500	12
3	Beta = 5	13
4	Lambda = 0.9	12
5	Beta= 5, Lambda = 0.05,	12
6	Reward, Lambda = 0.2	7

Table 18 shows marked changes in the rejected weight, which means that more good parts are rejected. In contrast, there is a slight increase in the rejected weight when the reward and lambda values are changed. The table also indicates that test 7 has the highest rejected weight where tests 1, 2, 4, and 5 have the same weight of 12%. Figure 25 depicts that the model's parameters have to be addressed carefully since any change in the parameters can change the results.

Table 19: RL-based Sequential Sampling vs. its Parameters' Comparisons

Test	Evaluation criteria	Sample rate	Bad accepted rate	Accepted weight	Bad rejected rate	Rejected weight
RL	**RL-based Seq. Sampling**	9	10	96	13	4
1	State thresholds constraints	12	10	88	15	12
2	Sample size: 8 to 50, Batch size = 500	22	10	88	15	12
3	Beta = 5	21	19	88	25	13
4	Lambda = 0.9	10	8	88	14	12
5	Beta= 5, Lambda = 0.05, Reward	49	16	88	22	12
6	Lambda = 0.2, Reward,	15	8	93	13	7

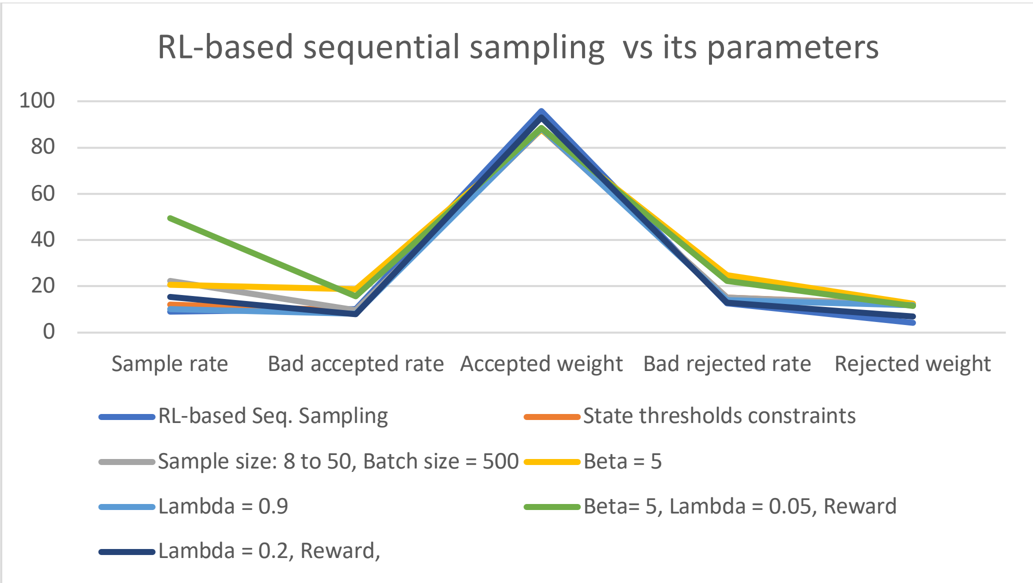


Figure 25 RL based Sequential Sampling vs. its Parameters

7.2.2 RL-Based Sequential Sampling vs. Acceptance Sampling Plans (Comparisons)

Part of studying the performance of the proposed model is to compare it with the performance of other acceptance sampling plans (existing sequential model, MIL-STD-1916 (32), MIL-STD-105E (Single sampling), MIL-STD-105E (Double sampling), and MIL-STD-105E (Multiple sampling). For simulating the acceptance sampling plans, we suppose that 100,000 products of part X are submitted in batches of size N = 500. The acceptable quality level (AQL) is 1.5% for a normal the single, double, and multiple sampling plans. Also, the normal verification level is considered for the inspection. The results of the comparisons obtained from 12 hours of running the algorithm for each of the simulated models are as shown in Table 20.

Table 20: Sample Rate-acceptance Sampling Plans

Evaluation criteria	Sample rate
RL-based Seq. Sampling	9
Sequential sampling	9
MIL-STD-1916 (32)	16
MIL-STD-105E (Single sampling)	10
MIL-STD-105E (Double sampling)	10
MIL-STD-105E (Multiple sampling)	10

Table 20 indicates that there are no major differences in sample rate for all the models.

However, MIL-STD-1916 appears to have the highest sample rate of 16%; that is because sometimes the sample can be increased when there are two or more batches are rejected right after each other. On the other hand, the sequential sampling plan has the same rate as the RL-based sequential sampling model with a 9% sample rate.

Table 21: Bad Accepted rate-acceptance Sampling Plans

Evaluation criteria	Bad accepted rate
RL-based Seq. Sampling	10
Sequential sampling	7
MIL-STD-1916 (32)	4
MIL-STD-105E (Single sampling)	6
MIL-STD-105E (Double sampling)	5
MIL-STD-105E (Multiple sampling)	4

The results in Table 21 indicate that RL-based sequential sampling and sequential sampling models have accepted more bad samples than the other models. That is because both models have a random sample size to inspect between a min of 8 and max of 50. Sometimes the sample size can be small and sometimes, it can be large, depending on how many samples an agent (inspector) wants to inspect.

Likewise, Table 22 shows that the accepted weight for both models is the highest, with 96% for RL-based sequential sampling and 63% for the sequential sampling model. By contrast, double sampling and multiple sampling have the lowest accepted weight. That is because the sample sizes selected for use in single, double, multiple plans are predetermined.

Table 22: Acceptance Weight-acceptance Sampling Plans

Evaluation criteria	**Accepted weight**
RL-based Seq. Sampling	96
Sequential sampling	63
MIL-STD-1916 (32)	49
MIL-STD-105E (Single sampling)	60
MIL-STD-105E (Double sampling)	53
MIL-STD-105E (Multiple sampling)	52

The results in Table 23 show that multiple sampling plans have the highest rate of rejecting a batch with a high number of bad samples. In contrast, RL and the sequential sampling model show the least rate of bad rejected rate. The reason is that the algorithm is trading off exploration for exploitation.

Table 23: Bad Rejected Rate-acceptance Sampling Plans

Evaluation criteria	**Bad rejected rate**
RL-based Seq. Sampling	13
Sequential sampling	16
MIL-STD-1916 (32)	16
MIL-STD-105E (Single sampling)	14
MIL-STD-105E (Double sampling)	14
MIL-STD-105E (Multiple sampling)	17

By contrast, Table 24 shows that there is a significant difference between the RL-based sequential sampling and the other models in rejected weight where the RL-based sequential sampling model only rejected 4% of the total sample size compared with 48% from multiple sampling.

Table 24: Rejected Weigh-acceptance Sampling Plans

Evaluation criteria	Rejected weight
RL-based Seq. Sampling	4
Sequential sampling	37
MIL-STD-1916 (32)	51
MIL-STD-105E (Single sampling)	40
MIL-STD-105E (Double sampling)	47
MIL-STD-105E (Multiple sampling)	48

Figure 26 depicts a comparison of the 6 models. It shows that the RL-based sequential sampling has the highest accepted weight and the lowest rejected weight. However, the rest of the rates are close to the other plans.

Table 25: Results of RL-based Sequential Sampling vs. Acceptance Sampling Plans

Evaluation criteria	Sample rate	Bad accepted rate	Accepted weight	Bad rejected rate	Rejected weight
RL-based Seq. Sampling	9	10	96	13	4
Sequential sampling	9	7	63	16	37
MIL-STD-1916 (32)	16	4	49	16	51
MIL-STD-105E (Single sampling)	10	6	60	14	40
MIL-STD-105E (Double sampling)	10	5	53	14	47
MIL-STD-105E (Multiple sampling)	10	4	52	17	48

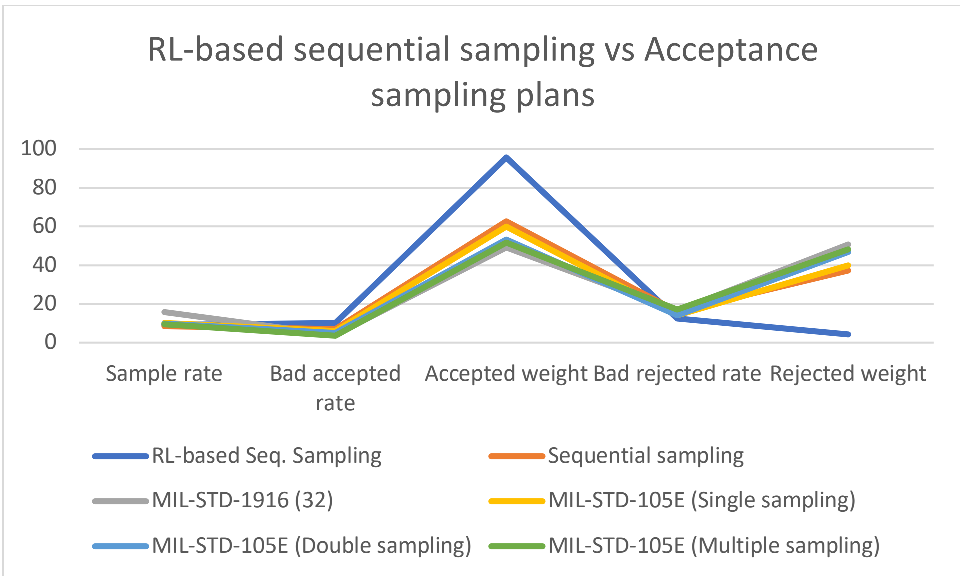


Figure 26: RL-based Sequential Sampling vs. Acceptance Sampling Plans

7.3 Results Summary and Discussion

Table 26 summarizes the results gathered from the various models of simulations. The data in the table are the sample rate, the bad accepted rate, the accepted weight, the bad rejected rate, and the rejected weight. The results in the table are used to compare the results of RL-based sequential sampling with the results of the simulated acceptance sampling plans and the results of different simulated parameters. For a complete visual representation of the comparison, histograms of these variables are included in the appendix.

Table 26: Summary of All Models in Percent

Test	Evaluation criteria	Sample rate	Bad accepted rate	Accepted weight	Bad rejected rate	Rejected weight
	RL-based Seq. Sampling	9	10	96	13	4
1	State thresholds constraints	12	10	88	15	12
2	Sample size: 8 to 50, Batch size = 500	22	10	88	15	12
3	Beta = 5	21	19	88	25	13
4	Lambda = 0.9	10	8	88	14	12
5	Beta= 5, Lambda = 0.05, Reward	49	16	88	22	12
6	Lambda = 0.2, Reward,	15	8	93	13	7
7	Sequential sampling	9	7	63	16	37
8	MIL-STD-1916 (32)	16	4	49	16	51
9	MIL-STD-105E (Single sampling)	10	6	60	14	40
10	MIL-STD-105E (Double sampling)	10	5	53	14	47
11	MIL-STD-105E (Multiple sampling)	10	4	52	17	48

Table 26 shows that all the model parameters used in the performance study have an impact on the implementation of RL-based sequential sampling. For example, a smaller number of lambda and beta could result in inspecting more samples; this can be seen in test 3, test 4, test 5, and test 6, where the sample rate is increased by 56%, 10%, 80%, and 60%, respectively. As a result, having a small value of lambda will not only increase the learning process of the agent but will also add more variability to the process.

Similarly, considering the small beta value in the process (test 3 and test 5) will increase the probability of sending a batch with a number of defectives that exceeds the rejectable quality level to the consumer (the bad accepted samples increases by 45% for test 3, and 35% for test 5). Unlike the small lambda (0.02) that results in a positive impact on the result of a bad accepted rate, test 4, where lambda is 0.9, resulted in a decrease in the bad accepted rate by 25%. Interestingly, there is an interaction effect whereby, when beta and lambda values are small and the reward parameters are increased, the bad accepted rate increased by 35%.

Lambda and beta can also change the performance of the model when it comes to the accepted weight. An increase in both values reduces the weight of acceptance by approximately 9% and that helps the agent to reject batches with a high number of defectives. Table 26 shows that all the considered scenarios have increased the rejected weight by at least 39%, except in test 6. When lambda is set to 0.2 and the rewards parameters are changed, test 6 results in the least rejected weight. Moreover, if a small beta value is added to the model, it will have the same impact on the sample rate as seen in test 3.

Table 24 also indicates that not only lambda and beta values can have an impact on the process and its results, but also the change in the state's thresholds constraint (test 1). This means that changing the state's thresholds from overlapping to having fixed boundaries (no overlap between state), has an impact in the results as seen in Figure 24. The same impact is realized when the sample size range is changed from 400 to 1500 (in the original model) to 8 to 50. In particular, the results from performing test 1 and test 2 show that both tests have the same impact on the bad accepted rate, the accepted weight, the bad rejected rate, and the rejected weight, but not on the sample rate. Table 26 shows that there is a 25% increase in the sample rate for test 1. Interestingly, the rejected weight increased by 65% compared with the main model, which means that the agent is limited to its transition space and takes more steps in each state before moving to another state.

Comparing acceptance sampling plans with the model explained earlier in Section 6.5, Table 26 shows an increase of the sample rate for all the acceptance except a sequential sampling plan. From the same table, using sequential sampling plan gives an advantage of having 7% less in the sample rate, 46% less in the bad accepted rate, and 21% more in the bad rejected rate. However, the sequential sampling plan's rejected weight is 89% more, which means it rejects more batches that contain not just defectives but also a lot of good parts. Thus, the rejected weigh for the sequential sampling plan is 89% higher than the proposed model.

In contrast, MIL-STD-1916 and MIL-STD-105E results show a marked difference compared with the result of RL-based sequential sampling. Both MIL standards show that they are rejecting good parts more than defectives. That is because both MIL-STD-1916 and MILE-STD-105E have fixed sample sizes. Additionally, MIL-STD-1196 follows a very strict policy, which is "zero defects," and that explains why 51% of the parts are rejected.

Chapter 8: Conclusions and Future Work

This chapter recapitulates this dissertation and concludes the work accomplished during this research. The main of objective of this dissertation is to integrate RL as a machine learning model to sequential acceptance sampling plan to minimize the number of sample sizes during the inspection process in a production line. In order to achieve the objective of the dissertation, the following sub-objectives have been accomplished:

1. The establishment of clear understanding and definitions of each of the following:
    a. The history of the industrial revolution, the manufacturing revolution sector and the history of the modern quality.
    b. The history of machine learning as a branch of artificial intelligence (supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning).
    c. Acceptance sampling plans such as single, double, multiple, and sequential sampling, and their current applications.
    d. Reinforcement Learning, its history, and its main components (agent, environment, policy, reward function, observation, and the action taken toward the environment), and its equations.
2. The development of straightforward formulations suited for use in explaining the process of sequential sampling using reinforcement learning.
3. The establishment of quality specifications and assumptions matching the purpose of initializing the simulation model.
4. The description of the constraints used for the simulation (min/max values for hidden batch size, sample size, wait time, and state thresholds).

5. The development of a flexible computer simulation program using Python programing for RL-based sequential sampling appropriate for implementation in a different acceptance sampling scenario.
6. The development of comparisons between the proposed model, sequential sampling, and MIL-STD 1916.

Based on the literature reviews and the results attained through this dissertation, the following statements may be made:

1. Sequential sampling is the most common plan that researchers have tried to improve due to its ability to reduce cost and sample size compared with the other acceptance sampling plans.
2. RL plays a major part for revolutionizing the industry of manufacturing.
3. Not many studies have been conducted to investigate the integration of RL into acceptance sampling.
4. Before considering any sampling methods, constraints for the problem must be considered. Having these constraints makes the algorithm more sample-efficient because they decrease the size of the search space.
5. Sequential acceptance sampling plays a major role to achieve the latest modern management principle objectives.
6. As a result of the above analysis and comparisons, RL-based sequential sampling has advantages over the other acceptance sampling plans, in that its sample rate is less, while detecting a similar proportion of more bad samples, so the risk of sending bad part to the costumers is minimal. It is also acceptant of relatively more good parts than the other acceptance plans that would rather penalize the process by rejecting more of the good parts.

Several promising research directions were identified throughout the development of this dissertation.

1. This dissertation has been focused on learning features in sequential acceptance sampling and reinforcement learning, by developing RL-based sequential sampling model and driving effective algorithms that can be applied not only in manufacturing but also to various sectors such as economics, medical, and any other large scale sectors. The contribution of this research is to solve the sequential acceptance sampling optimization problem using the RL methodology, by reducing the inspection sample size and wait time, and thus reducing the inspection costs.
2. There is more work to be done to test whether the algorithm can respond to a heterogeneous rate of non-conformance, i.e., when the process is not Poisson in nature.
3. More study can be accomplished on how to evaluate the importance and impact of the constraints in the algorithm.
4. More studies can be made in the future to investigate the effectiveness and efficiencies of the proposed model on a real-world problem. In practice, first more analysis is needed to define the optimal constraints' parameters.
5. More analysis into the state transition cost can be done by adding to the complexity of the transition cost calculation.
6. It was noted earlier that the model parameters promote agent exploration rather than exploitation, which we suspect is impacting the number of detected defective parts in the samples. Further research is needed to fine-tune model parameters to enable quick initial exploration followed by exploitation to avoid the trade-off being made between lowering the sampling rate and increasing the bad rejected rate.

7. More studies can be made on whether or not sequential sampling with RL can be considered as a critical success for lean six sigma (LSS), the latest modern quality.

As testing and implementation of the RL-based sequential sampling model, more questions will be answered, and more recommendations will be made in the future. The background of acceptance sampling, RL, and the methods in this dissertation will serve as a first step for further improvements in all the areas mentioned in the work.

References

Afshari, R., & Gildeh, B. (2017). Modified sequential sampling plan using fuzzy SPRT. In *2017 5th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)* (pp. 7–9). Tehran.

Albliwi, S., Antony, J., & Lim, S. (2015). A systematic review of lean six sigma for the manufacturing industry. *Business Process Management Journal*, *21*(3), 665–691. https://doi.org/10.1108/BPMJ-03-2014-0019

Alhuraish, I., Robledo, C., & Kobi, A. (2017). A comparative exploration of lean manufacturing and six sigma in terms of their critical success factors. *Journal of Cleaner Production*, *164*, 325–337. https://doi.org/10.1016/j.jclepro.2017.06.146

Allen, T. T. (2010). Statistical Quality Control and Six Sigma. *Introduction to Engineering Statistics and Lean Sigma: Statistical Quality Control and Design of Experiments and Systems*, 31–46.

Aslam, Muhammad; Khan, Nasrullah; and Khan, H. (2015). SkSP-V Acceptance sampling plan based on process capability Index. *Chiang Mai J. Sci.*, *42*(1), 258–267. Retrieved from http://epg.science.cmu.ac.th/ejournal/

ASQ. (2019). What is Total quality management (TQM)?

Azizzadenesheli, K., Lazaric, A., & Anandkumar, A. (2017). Experimental results : Reinforcement Learning of POMDPs using Spectral Methods, (element 1), 1–9. Retrieved from http://arxiv.org/abs/1705.02553

Barone, S., & Eva, L. F. (2012). *Statistical and managerial techniques for six sigma methodology: theory and application*. Chichester, UK: John Wiley & Sons, Ltd.

Borror, C. M. (2009). *Certified quality engineer handbook* (Third). Delhi: American Society for Quality (ASQ).

Chen, D., Heyer, S., Ibbotson, S., Salonitis, K., Steingrímsson, J. G., & Thiede, S. (2015). Direct digital manufacturing: Definition, evolution, and sustainability implications. *Journal of Cleaner Production*, *107*, 615–625. https://doi.org/10.1016/j.jclepro.2015.05.009

Chen, K. (2014). *Scheduling mass customized large product assembly line considering learning effect and shifting bottleneck*. Retrieved from https://search.proquest.com/docview/1528568992/?pq-origsite=primo

Chetouani, Y. (2013). A sequential probability ratio test (SPRT) to detect changes and process safety monitoring. *Process Safety and Environmental Protection*, *92*(3), 206–214. https://doi.org/10.1016/j.psep.2013.02.001

Chick, S. E., Forster, M., & Pertile, P. (2015). Optimal sequential sampling with delayed observations and unknown variance. In *Proceedings of the 2015 Winter Simulation Conference* (Vol. 2016-Febru). IEEE. https://doi.org/10.1109/WSC.2015.7408536

Childs, A., & Chen, Y. (2011). Multilevel Fixed and Sequential Acceptance Sampling: The R Package MFSAS. *Journal of Statistical Software*, *43*(6). https://doi.org/http://dx.doi.org/10.18637/jss.v043.i06

Cudney, E. A., Qin, R., & Hamzic, Z. (2016). Development of an optimization model to determine sampling levels. *International Journal of Quality & Reliability Management*, *33*(4), 476–487. https://doi.org/10.1108/IJQRM-10-2014-0159

Dornheim, J., Link, N., & Gumbsch, P. (2018). Model-Free Adaptive Optimal Control of Sequential Manufacturing Processes using Reinforcement Learning. *Cornell University*, 1–13. Retrieved from http://arxiv.org/abs/1809.06646

Ellis, S. F. (2016). *The application of lean six sigma to improve a business process: A study of the order processing process at an automobile manufacturing facility*. Retrieved from https://ezproxy.lib.uwm.edu/login?url=https://search.proquest.com/docview/187457564

5?accountid=15078
Fallahnezhad, M., & Babadi, A. (2016). Cost analysis of acceptance sampling models using dynamic programming and bayesian inference considering inspection errors. *Journal of Optimization in Industrial Engineering*, *19*, 9–24.
Fallahnezhad, M., Babadi, A., Momeni, M., Sayani, N., & Akhoondi, F. (2015). Optimal Sequential Sampling Plans Using Dynamic Programming Approach. *Pakistan Journal of Statistics and Operation Research*, (4), 575–586.
Fathi, M., Maihami, V., & Moradi, P. (2013). Reinforcement learning for multiple access control in wireless sensor networks: Review, model, and open issues. *Wireless Personal Communications*, *72*(1), 535–547. https://doi.org/10.1007/s11277-013-1028-9
Feigenbaum, A. V. (1991). *Total quality control (Third edition, revised. ed.)*. New York: McGraw-Hill.
Feigenbaum, A. V. (2019). DR. ARMAND V. FEIGENBAUM. Retrieved from http://www.feigenbaumfoundation.org/about/dr-armand-v-feigenbaum/
Fernández, A. J. (2015). Optimal schemes for resubmitted lot acceptance using previous defect count data. *Computers & Industrial Engineering*, *87*, 66–73. https://doi.org/10.1016/j.cie.2015.04.015
Franchetti, M. J. (2015). *Lean six sigma for engineers and managemers*. Boca Raton: FLTaylor & Francis Group.
Fudenberg, D., Strack, P., & Strzalecki, T. (2015). Stochastic choice and optimal sequential sampling ∗. *ASSA Annual Meetings*, 1–38.
Gatti, C. (2015). Design of Experiments for Reinforcement Learning, *2014*(August). https://doi.org/10.1007/978-3-319-12197-0
George, M. L. (2002). *Lean six sigma: combining six sigma quality with lean production speed*. New York: The McGraw-Hill Companies.
Gray, A. (2017). Agenda. Retrieved from https://www.weforum.org/agenda/2017/03/worlds-biggest-economies-in-2017/
Griffin, P. M., Nembhard, H. B., Deflitch, C. J., Bastian, N. D., Kang, H., & Muñoz, D. A. (2016). *Healthcare systems engineering*. Hoboken, NJ, USA: John Wiley & Sons, Incorporated.
Hamzic, S. (2013). *Development of an optimization model to determine sampling levels*. Retrieved from ttps://ezproxy.lib.uwm.edu/login?url=https://search.proquest.com/docview/1507845031?accountid=15078
Hartman, P. L., Ogden, J. A., Wirthlin, J. R., & Hazen, B. T. (2017). Nearshoring , reshoring , and insourcing : Moving beyond the total cost of ownership conversation. *Business Horizons*, *60*(3), 363–373. https://doi.org/10.1016/j.bushor.2017.01.008
Herwan, J., Kano, S., Oleg, R., Sawada, H., & Kasashima, N. (2018). Cyber-physical system architecture for machining production line. *Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018*, 387–391. https://doi.org/10.1109/ICPHYS.2018.8387689
Hu, S. J. (2013). Forty-sixth cirp conference on manufacturing systems (cirp cms 2013): economic development and wealth through globally competitive manufacturing systems. In P. F. Cunha (Ed.), *CIRP Volume 7*. Setubal, Portugal.
Hudson, P. (2014). *The industrial revolution*. New York: Bloomsbury Publishing Plc.
Jamkhaneh, E., & Gildeh, B. (2013). Sequential sampling plan using fuzzy SPRT. *Journal of Intelligent & Fuzzy Systems 25*, *25*, 785–791. https://doi.org/10.3233/IFS-120684
Jones, E. C. (2014). *Quality management for organizations using Lean Six Sigma techniques*. Boca Raton, FL: Taylor & Francis Group, LLC.
Kara, A., & Dogan, I. (2018). Reinforcement learning approaches for specifying ordering policies of perishable inventory systems. *Expert Systems with Applications*, *91*, 150–

158. https://doi.org/10.1016/j.eswa.2017.08.046
Katanyukul, T., & K. P. Chong, E. (2014). Intelligent inventory control via ruminative reinforcement learning. *Journal of Applied Mathematics*, *2014*. Retrieved from http://dx.doi.org/10.1155/2014/238357%0AResearch
Koren, Y. (2010). *The global manufacturing revolution product-process-business integration and reconfigurable systems*. Hoboken, NY: John Wiley & Sons, Inc. Retrieved from Willy
Kulkarni, P. (2012). *Reinforcement and systemic machine learning decision making*. Hoboken, NJ, USA: John Wiley & Sons.
Kumar, M., & P. C., R. (2016). Design of optimal reliability acceptance sampling plans for exponential distribution. *Economic Quality Control*, *31*(1), 23–36. https://doi.org/10.1515/eqc-2015-0005
Lal, B., Kumar, P., & Bhardwaj, D. (2014). A Study and Analysis Six-Sigma Methodologies in Manufacturing Industries. *Certified International Journal of Engineering and Innovative Technology*, *9001*(8), 2277–3754.
Lee, J. H., Shin, J., & Realff, M. J. (2018). Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers and Chemical Engineering*, *114*, 111–121. https://doi.org/10.1016/j.compchemeng.2017.10.008
Li, H., Cai, R., Liu, N., Lin, X., & Wang, Y. (2018). Deep reinforcement learning: Algorithm, applications, and ultra-low-power implementation. *Nano Communication Networks*, *16*, 81–90. https://doi.org/10.1016/j.nancom.2018.02.003
Li, M. H. C., & Tsao, C. W. (2011). A Study on the Attributes Sampling Plans in MIL-STD-1916, *I*.
Liker, J. K. (2004). *The Toyota way: 14 management principles from the world's greatest manufacturer* (Vol. 2004). New York: McGraw-Hill. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract
Mannion, P., Devlin, S., Mason, K., Duggan, J., & Howley, E. (2017). Policy invariance under reward transformations for multi-objective reinforcement learning. *Neurocomputing*, *263*, 60–73. https://doi.org/10.1016/j.neucom.2017.05.090
MIL-HDBK-1916. (1999). *Department of defense handbook: companion document to MIL-STD-1916*.
MIL-STD-105E. (1989). *Military standard: sampling procedures and tables for inspection by attributes*.
MIL-STD-1916. (1996). *Department of defense test method standard DOD preferred methods for acceptance of product*.
Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., … Welling, J. (2018). Never-ending learning. *Communications Of The Acm*, 1–4. https://doi.org/10.1007/s13398-014-0173-7.2
Mitra, A. (2016). *Fundamentals of quality control and improvement* (4th ed.). Hoboken, NJ, USA: John Wiley & Sons, Inc.
Montgomery, D. C. (2013). *Introduction to statistical quality control (seven Ed)*. Hoboken: John Wiley & Sons, Inc.
Naidu, N., Babu, K. M., & Rajendra, G. (2006). *Total quality management*. New Delhi: New Age International (P) Ltd.
Nassima, A., Bouziane, B., & Amine, M. (2013). Heurestics and reinforcement learning in manufacturing control : optimization by phases. *IEEE Access*, 798–803.
Oddi, G., & Pietrabissa, A. (2013). A distributed multi-path algorithm for wireless ad-hoc networks based on Wardrop routing. In *21st Mediterranean Conference on Control and Automation* (2nd editio, Vol. 3, pp. 930–935). London: IEEE.

https://doi.org/10.1109/MED.2013.6608833
Otieno, W., & Nanduri, V. (2012). An attribute quality sampling strategy using a reinforcement learning methodology. In *The 3rd International Conference on Technology and Operations Management "Sustaining Competitiveness through Green Technology Management"* (pp. 978–979). Bandung.
Ou, X., Chang, Q., Arinez, J., & Zou, J. (2018). Gantry work cell scheduling through reinforcement learning with knowledge-guided reward setting. *IEEE Access*, *6*, 14699–14709. https://doi.org/10.1109/ACCESS.2018.2800641
Pathak, V., & Tiwari, P. (2018). *Artificial intelligence for all: an abiding destination*. Dwarka, New Delhi: Educreation publishing.
Pepper, M. P. J., & Spedding, T. A. (2010). The evolution of lean Six Sigma. *International Journal of Quality & Reliability Management*, *27*(2), 138–155. https://doi.org/10.1108/02656711011014276
Régio, M. M. de A., Gaspar, M. R. C., Farinha, L. M. do C., & Morgado, M. M. A. de P. (2016). Forecasting the disruptive skillset alignment induced by the forthcoming industrial revolution. *Romanian Review Precision Mechanics, Optics and Mechatronics*, *2016*(49), 24–29. https://doi.org/http://dx.doi.org/10.17683/rrpmom.issue.49
Schilling, E. G., & Neubauer, D. V. (2017). *Acceptance sampling in quality control* (Third Edit). Boka Raton: Taylor & Francis Group,LLC.
Schwab, K. (2016). *No The fourth industrial revolution*. New Yorl, USA: Crown Publishing Group.
Shadlen, M. N., & Shohamy, D. (2016). Decision making and sequential sampling from memory. *Neuron Perspective*, *90*(5), 927–939. https://doi.org/10.1016/j.neuron.2016.04.036
Shahrabi, J., Adibi, M. A., & Mahootchi, M. (2017). A reinforcement learning approach to parameter estimation in dynamic job shop scheduling. *Computers and Industrial Engineering*, *110*, 75–82. https://doi.org/10.1016/j.cie.2017.05.026
Si, J., Barto, A., Powell, W., & Wunsch, D. (2012). Reinforcement learning and its relationship to supervised learning. In *Handbook of Learning and Approximate Dynamic Programming* (pp. 45–63). Hoboken: John Wiley & Sons. https://doi.org/10.1002/9780470544785.ch2
Silver, D. (2015). UCL course on RL. Lecture 1: introduction to reinforcement learning. Retrieved from http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
Smalley, A. (2013). Toyota production system basic handbook. Retrieved from www.artoflean.com
Sridharan, A. R. (2015). *An approach to inventory management in mass customized printing production environment*. Retrieved from https://ezproxy.lib.uwm.edu/login?url=https://search.proquest.com/docview/1735757733?accountid=15078
Starvaggi, P. W. (2014). *Exact Distributions of Sequential Probability Ratio Test*. Kent State University. Retrieved from http://rave.ohiolink.edu/etdc/view?acc_num=kent1397042380
Stecher, B., & Kirby, S. N. (2004). *Organizational Improvement and Accountability: Lessons for Education from Other Sectors*. Santa Monica, CA; Arlington, VA; Pittsburgh: RAND Corporation. Retrieved from http://www.jstor.org/stable/10.7249/mg136wfhf
Stricker, N., Kuhnle, A., Sturm, R., & Friess, S. (2018). Reinforcement learning for adaptive order dispatching in the semiconductor industry. *CIRP Annals-Manufacturing Technology*, *67*(1), 511–514. https://doi.org/10.1016/j.cirp.2018.04.041
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: an introduction* (2nd Editio). London: A Bradford Book.

Syam, N., & Sharma, A. (2018). Waiting for a sales renaissance in the fourth industrial revolution : Machine learning and artificial intelligence in sales research and practice. *Industrial Marketing Management*, *69*(December 2017), 135–146. https://doi.org/10.1016/j.indmarman.2017.12.019

Taylor, J. (2014). *Development of an optimal Lean Six Sigma model*.

The ISO story. (2017). Retrieved from https://www.iso.org/the-iso-story.html

Thompson, C. B. (1917). *The theory and practice of scientific management*. Boston: Houghton Mifflin.

Trew, A. (2014). Spatial takeoff in the first industrial revolution. *Review of Economic Dynamics*, *17*(4), 707–725. https://doi.org/10.1016/j.red.2014.01.002

Walker, H. F., Elshennawy, A. K., Gupta, B. C., & McShane Vaughn, M. (2012). *Certified quality inspector handbook* (2nd ed.). Milwaukee: American Society for Quality (ASQ).

Wood, J., Thuy, N., Sharda, S., Scholsser, A., Menon, J., Fink, A., … Tey, S. (2017). ASEAN 4.0: What does the Fourth Industrial Revolution mean for regional economic integration? *World Economic Forum*, (November). https://doi.org/http://dx.doi.org/10.22617/TCS179126-2

World Economic Forum. (2016). Manufacturing Our Future Cases on the Future of Manufacturing. *World Economic Forum*, (May). Retrieved from http://www3.weforum.org/docs/GAC16_The_Future_of_Manufacturing_report.pdf

Wu, C.-W., & Liu, S.-W. (2014). Developing a sampling plan by variables inspection for controlling lot fraction of defectives. *Applied Mathematical Modelling*, *38*(9–10), 2303–2310. https://doi.org/10.1016/j.apm.2013.10.043

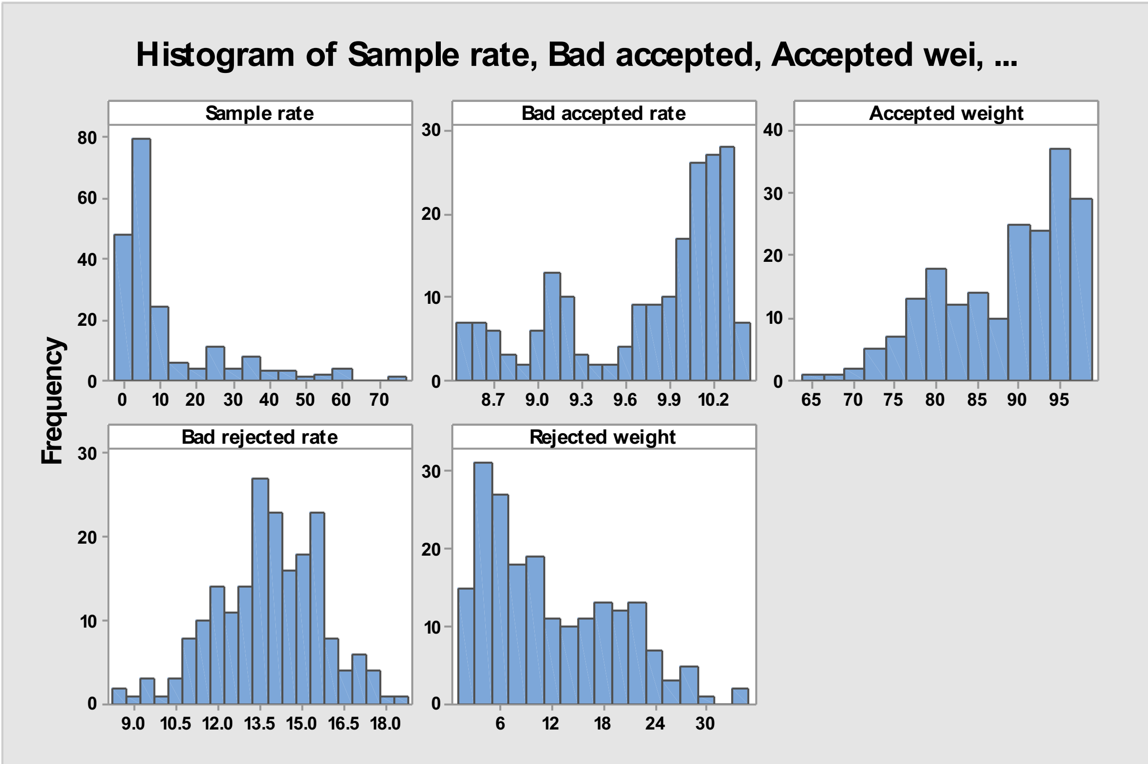
Zairi, M. (1991). *Total quality management for engineers*. Cambridge: Woodhead Publishing Ltd.

Zarandi, M., Moosavi, H., & Zarinbal, F. (2013). A fuzzy reinforcement learning algorithm for inventory control in supply chains. *The International Journal of Advanced Manufacturing Technology*, 557–569. https://doi.org/10.1007/s00170-012-4195-z

APPENDICES

Appendix A: RL-Based Sequential Sampling Simulation

APPENDIX A1: RL-Based Sequential Sampling Histogram



APPENDIX A2: RL-Based Sequential Sampling Figures

Figure A2-1: RL- based sequential sampling agent's behavior for sample rate

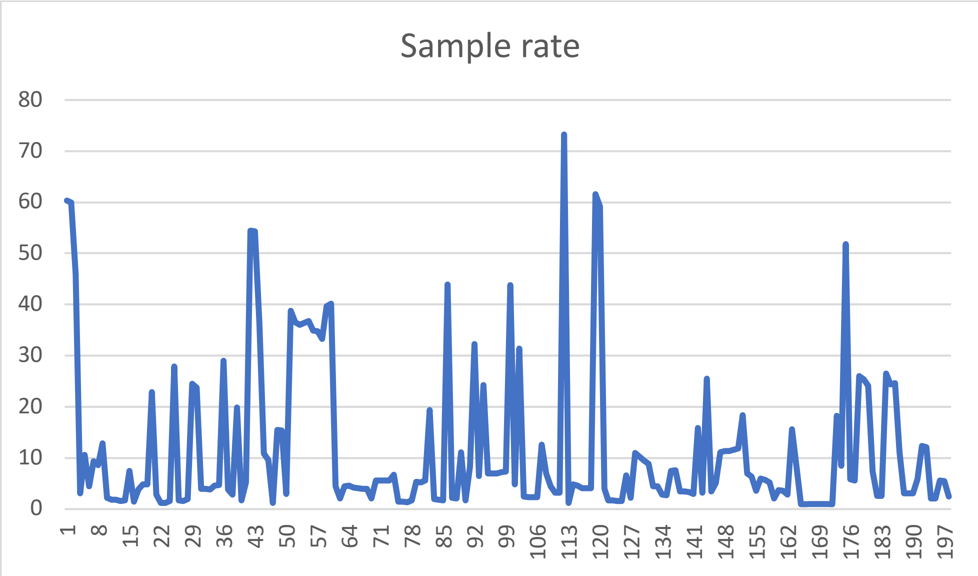


Figure A2-2: RL- based sequential sampling agent's behavior for bad accepted rate

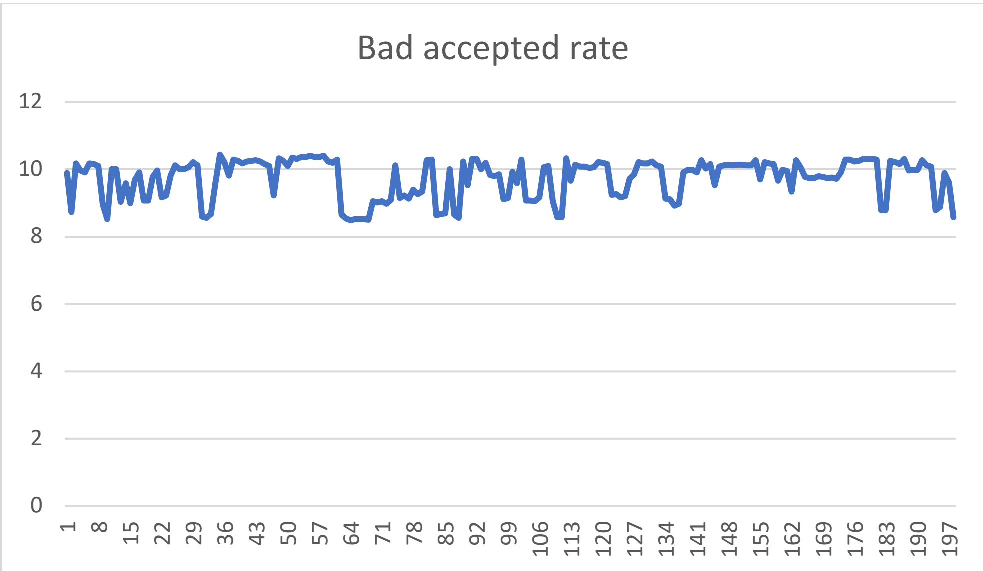


Figure A2-3: RL- based sequential sampling agent’s behavior for accepted weight

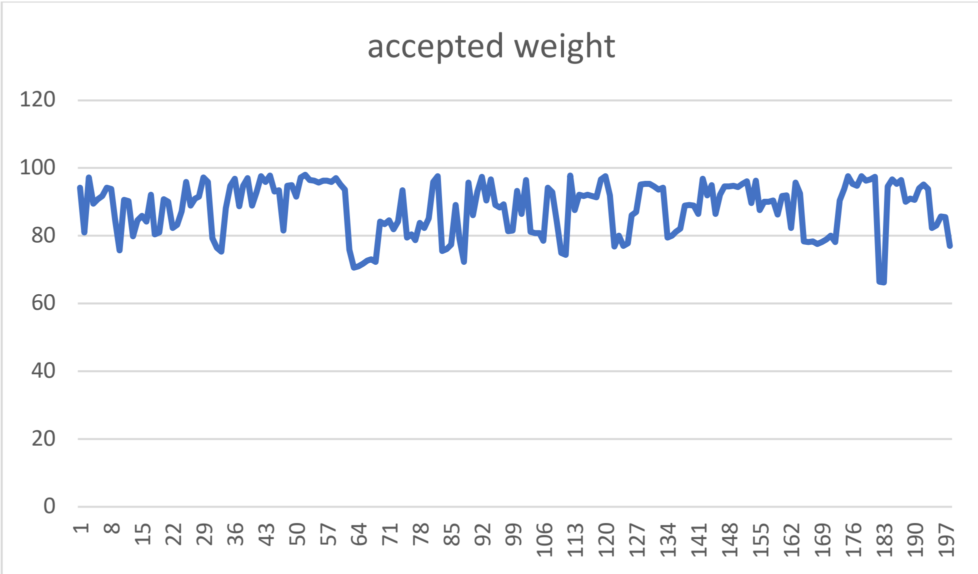


Figure A2-4: RL- based sequential sampling agent’s behavior for bad rejected rate

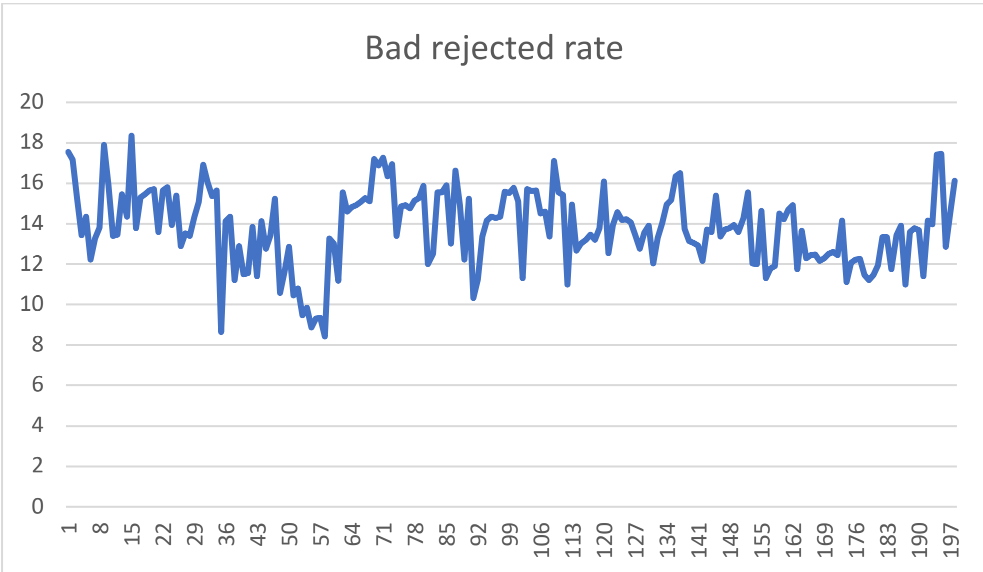


Figure A2-5: RL- based sequential sampling agent's behavior for rejected weight

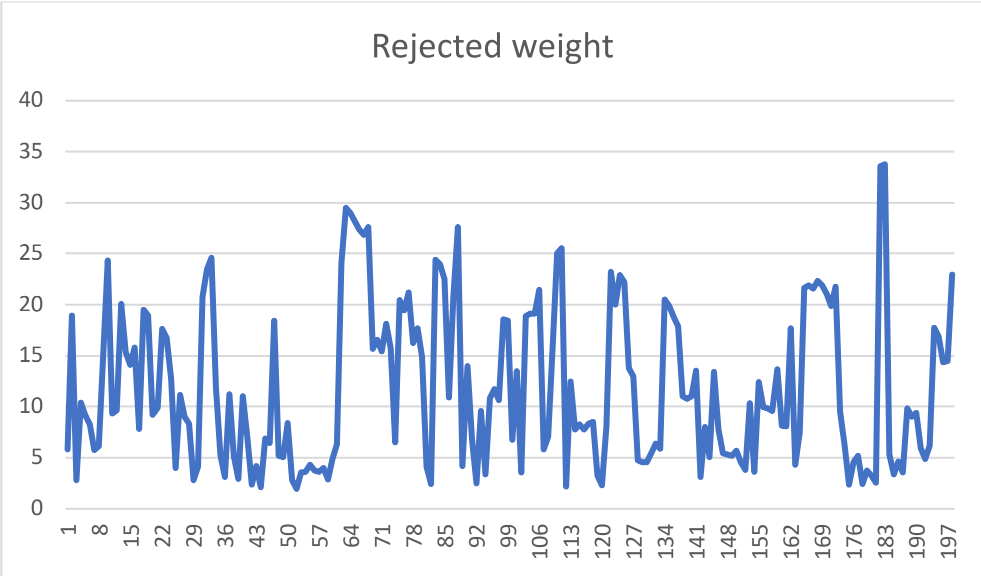
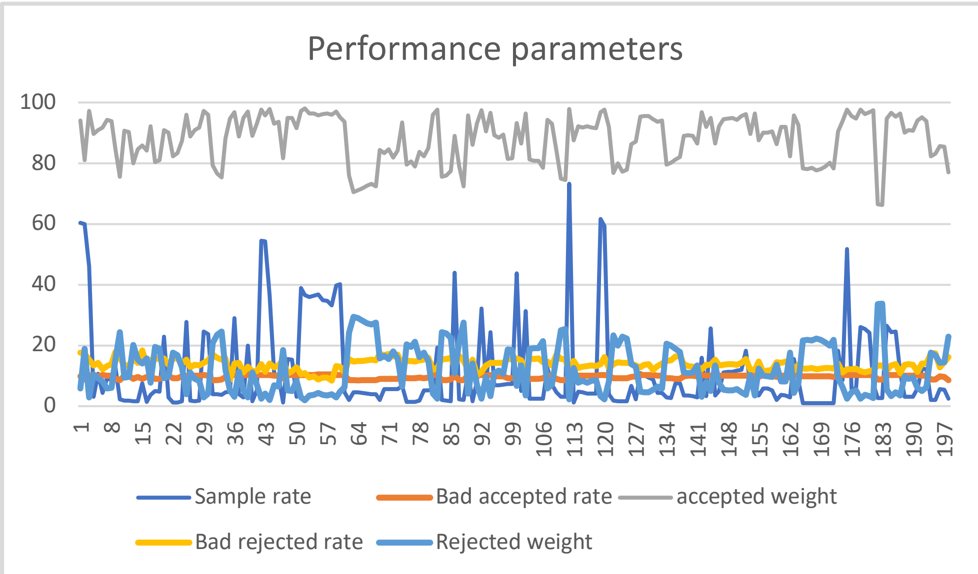
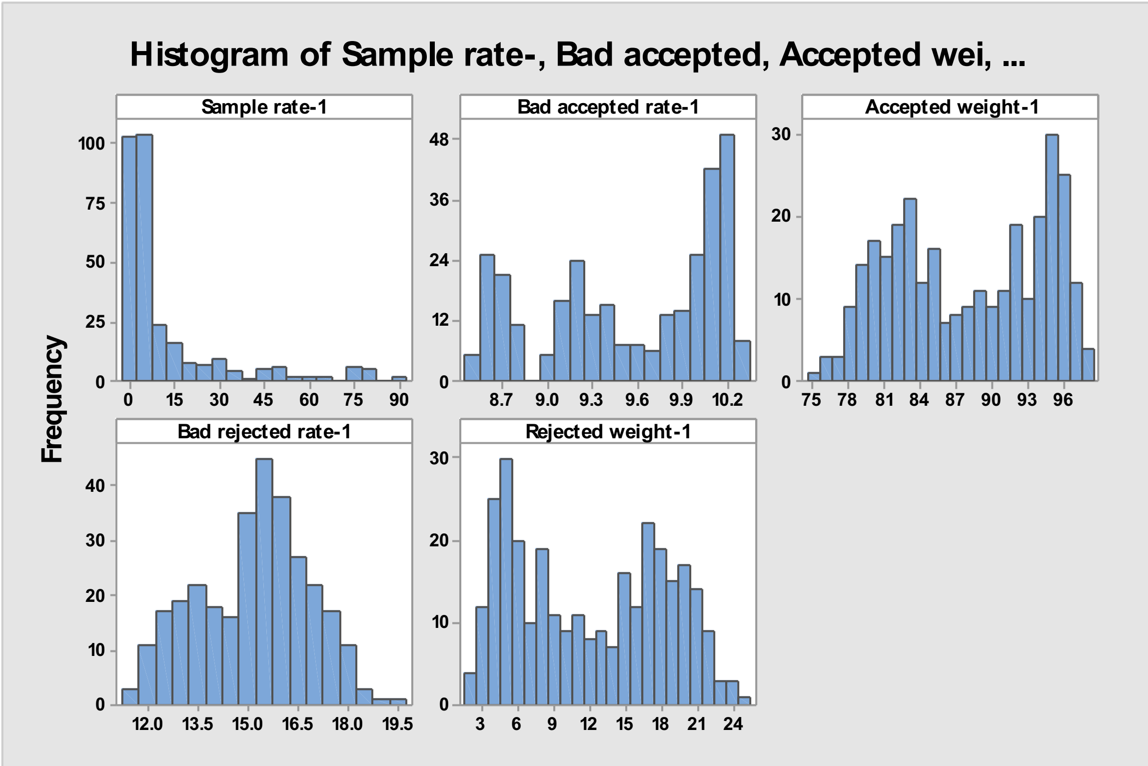


Figure A2-6: RL- based sequential sampling agent's behavior for all performance parameters



APPENDIX B: RL-Based Sequential Sampling Simulation Histogram for State Thresholds Constraints (Test 1)

APPENDIX B: RL-Based sequential sampling Histogram for state thresholds constraints (test 1)



APPENDIX B2: RL-Based Sequential Sampling Figures for State Thresholds Constraints (Test 1)

Figure B2-1: RL- based sequential sampling agent’s behavior for sample rate (Test1)

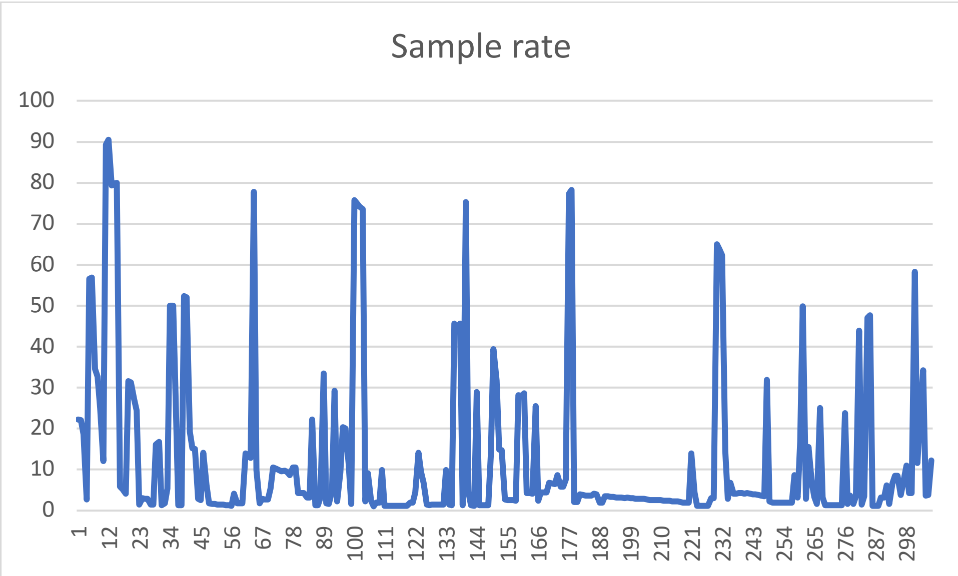


Figure B2-2: RL- based sequential sampling agent’s behavior for bad accepted rate(Test1)

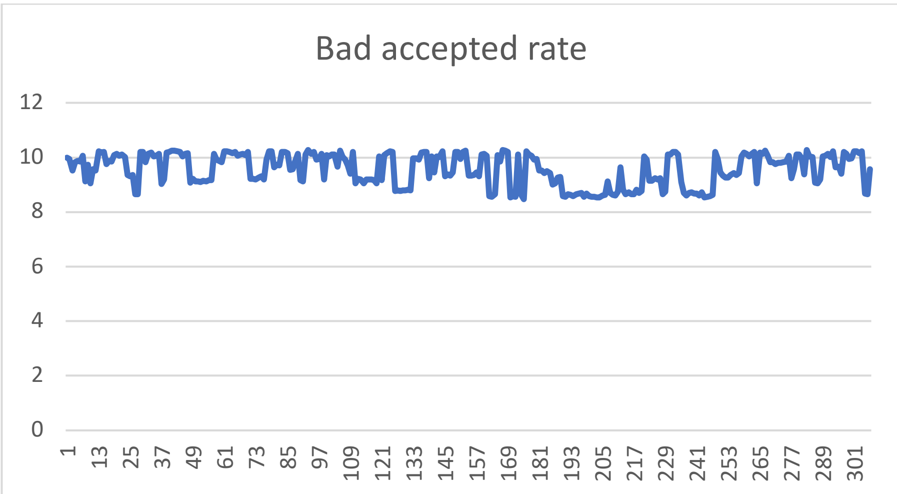


Figure B2-3: RL- based sequential sampling agent’s behavior for accepted weight (Test1)

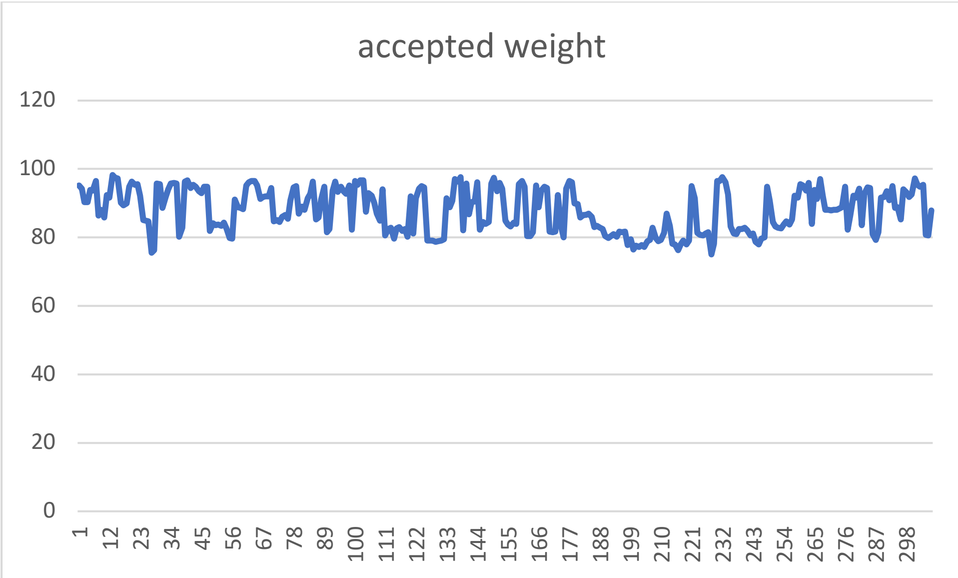


Figure B2-4: RL- based sequential sampling agent’s behavior for bad rejected rate (Test1)

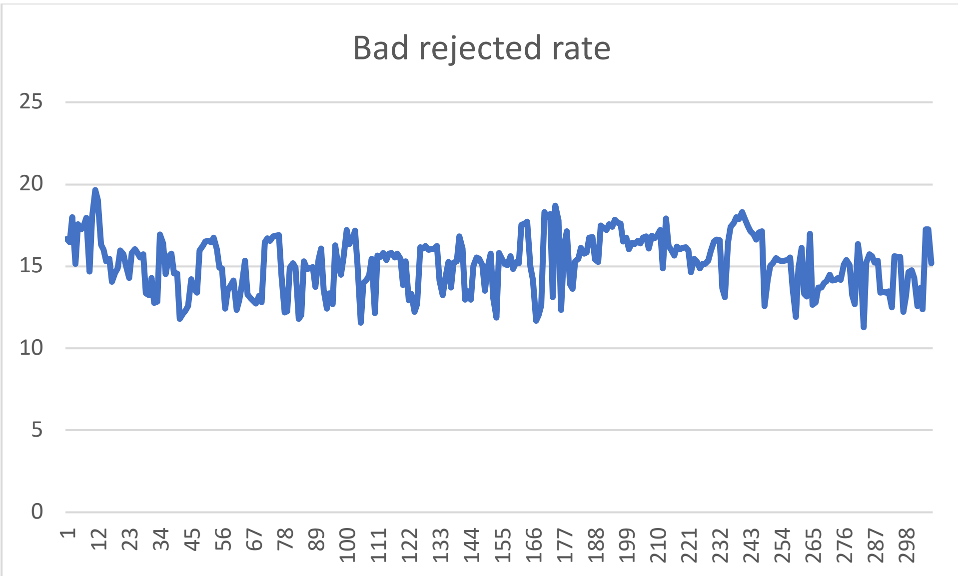


Figure B2-5: RL- based sequential sampling agent's behavior for rejected weight (Test1)

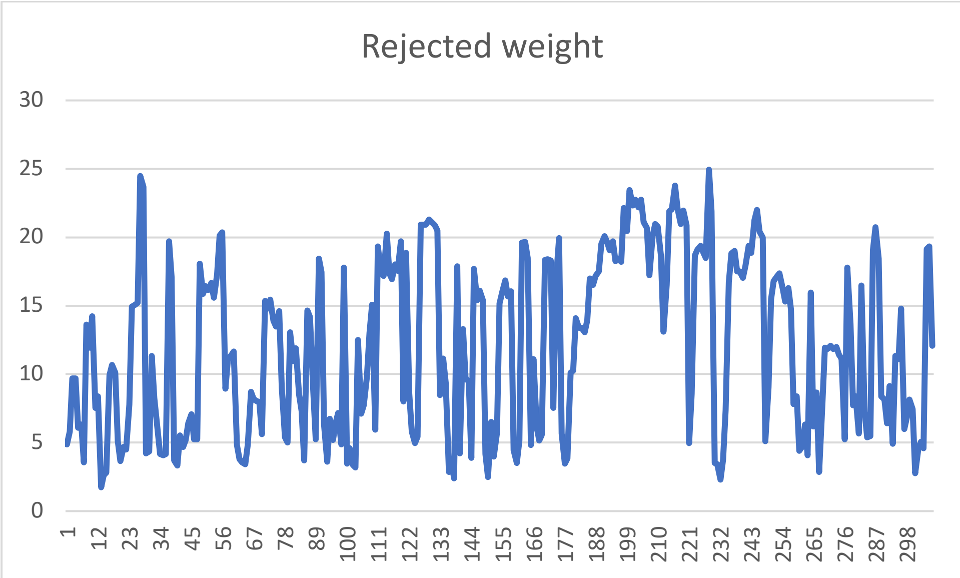
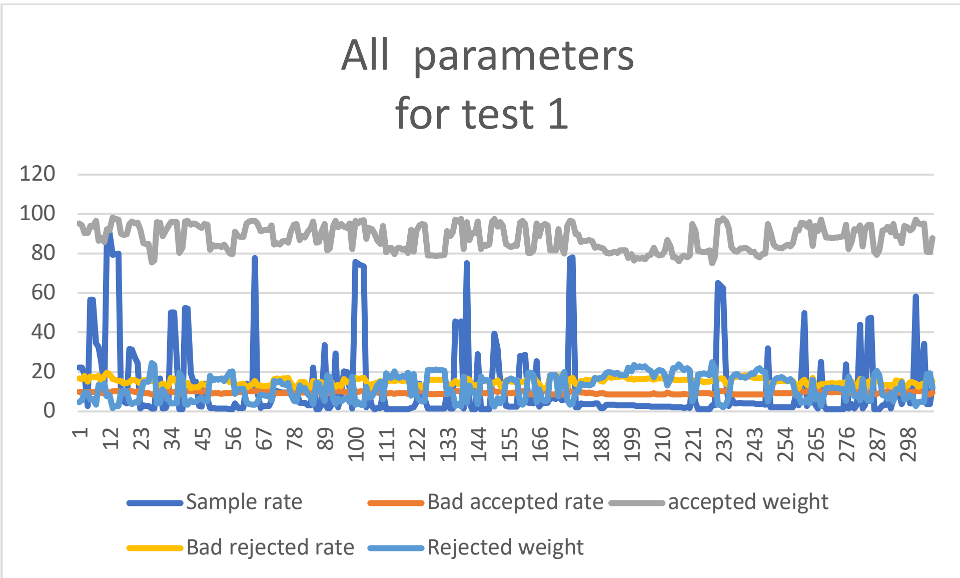
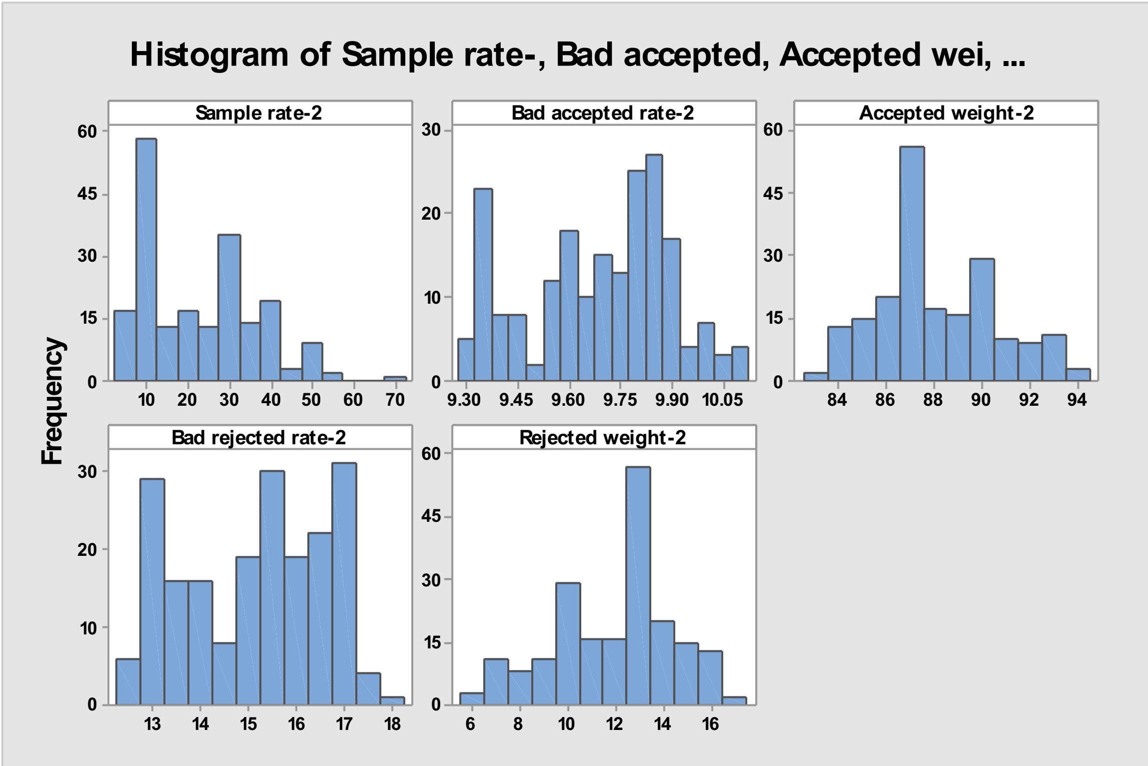


Figure B2-6: RL- based sequential sampling agent's behavior for sample rate (Test1)



APPENDIX C: RL-Based Sequential Sampling Simulation Histogram for Sample Size: 8 To 50, Batch Size = 500 (Test 2)

APPENDIX C: RL-Based sequential sampling Histogram for Sample size: 8 to 50, Batch size = 500 (test 2)



APPENDIX C2: RL-Based Sequential Sampling Figures for Sample Size: 8 To 50, And Batch Size = 500 (Test 2)

Figure C2-1: RL- based sequential sampling agent's behavior for sample rate (Test 4)

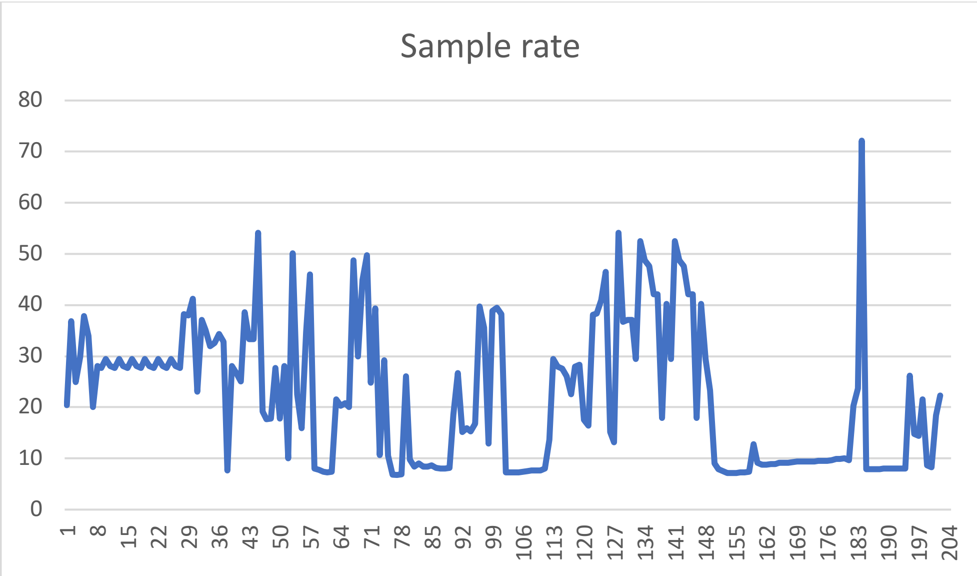


Figure C2-2: RL- based sequential sampling agent's behavior for bad accepted rate (Test 2)

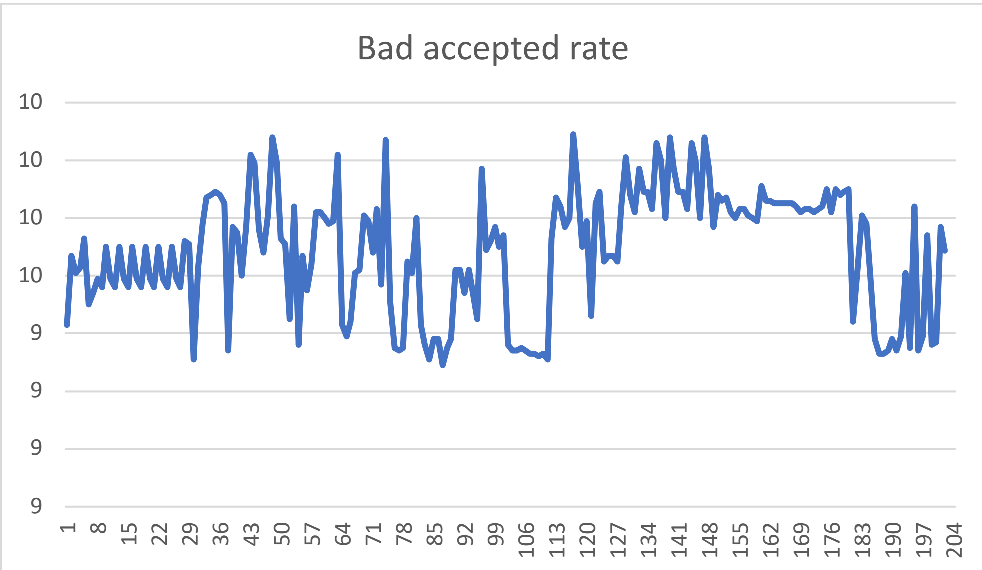


Figure C2-3: RL- based sequential sampling agent's behavior for accepted weight (Test 2)

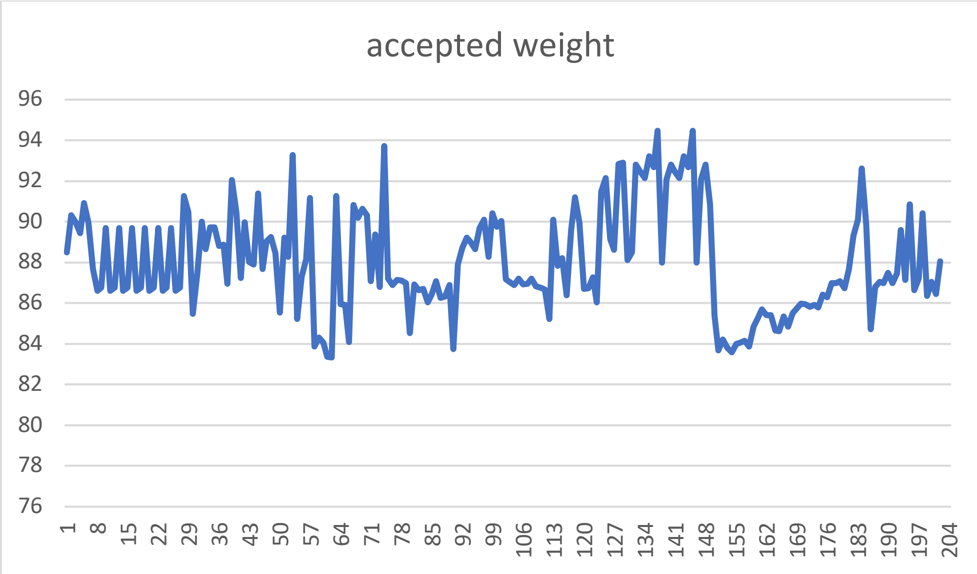


Figure C2-4: RL- based sequential sampling agent's behavior for bad rejected rate (Test 2)

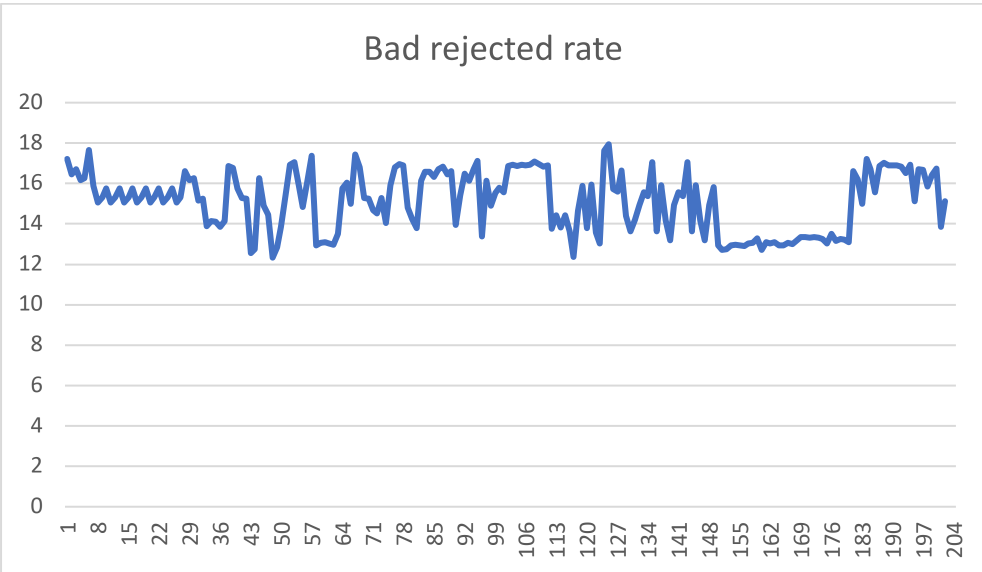


Figure C2-5: RL- based sequential sampling agent's behavior for rejected weight (Test 2)

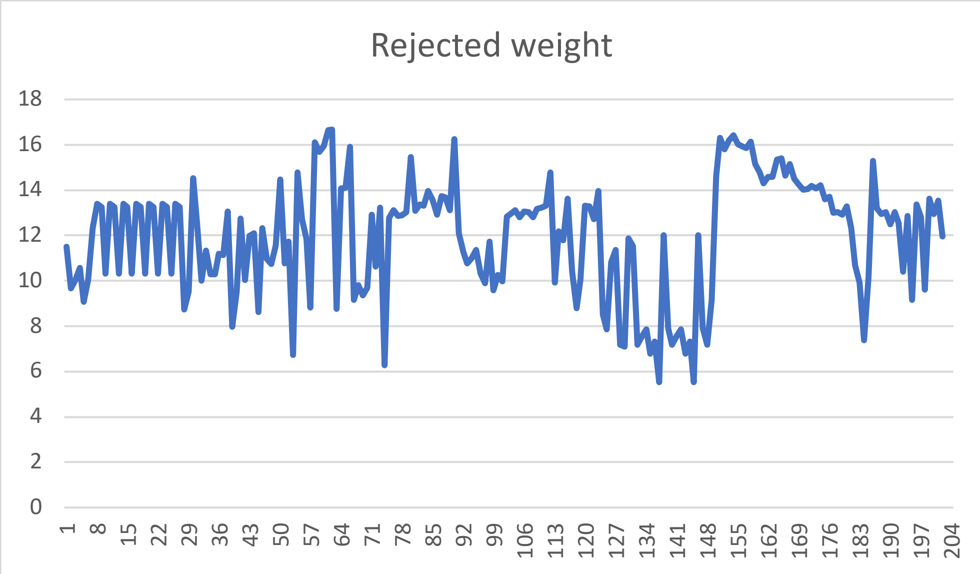
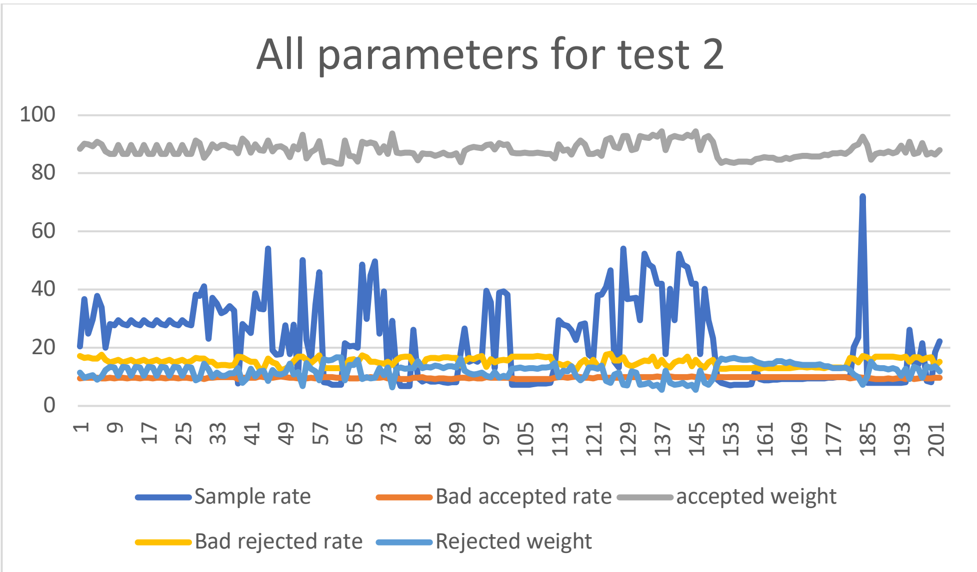
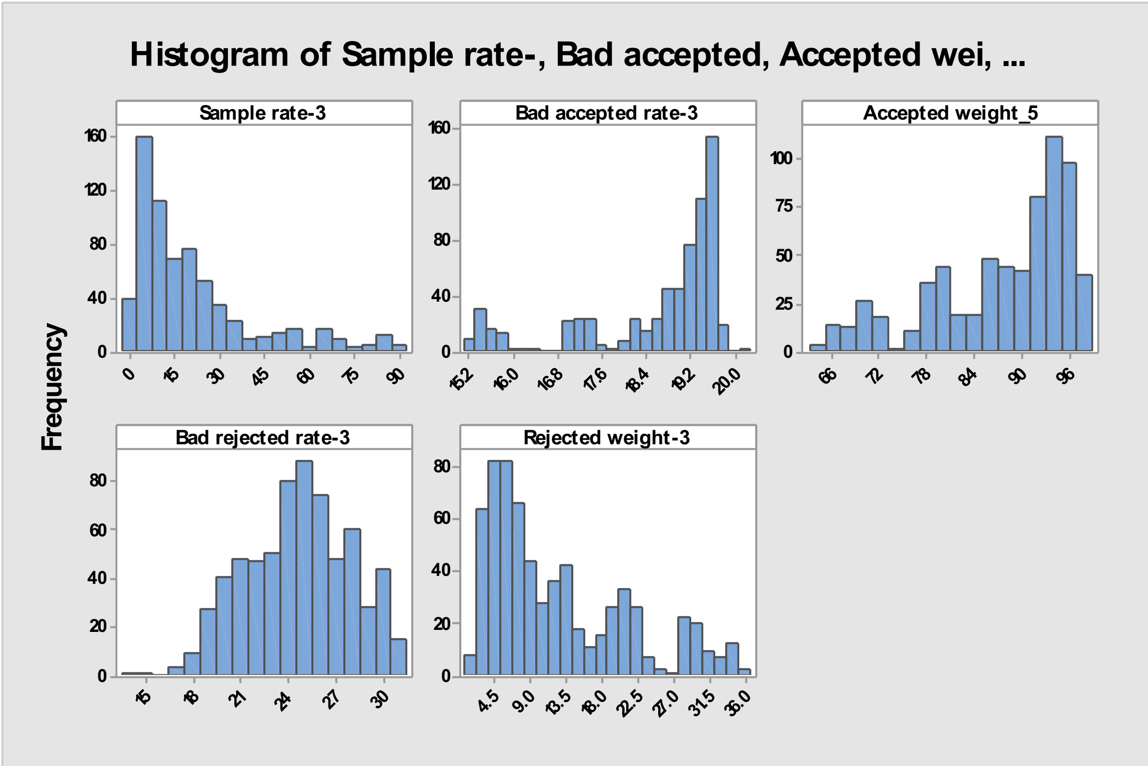


Figure C2-6: RL- based sequential sampling agent's behavior for all parameters (Test 2)



APPENDIX D: RL-Based Sequential Sampling Simulation Histogram for Beta = 5 (Test 3)

APPENDIX D: RL-Based sequential sampling Histogram for Beta = 5



APPENDIX D2: RL-Based Sequential Sampling Figures for Sample Size: Beta= 5 (Test 3)

Figure D2-1: RL- based sequential sampling agent's behavior for Beta =5 (Test 4)

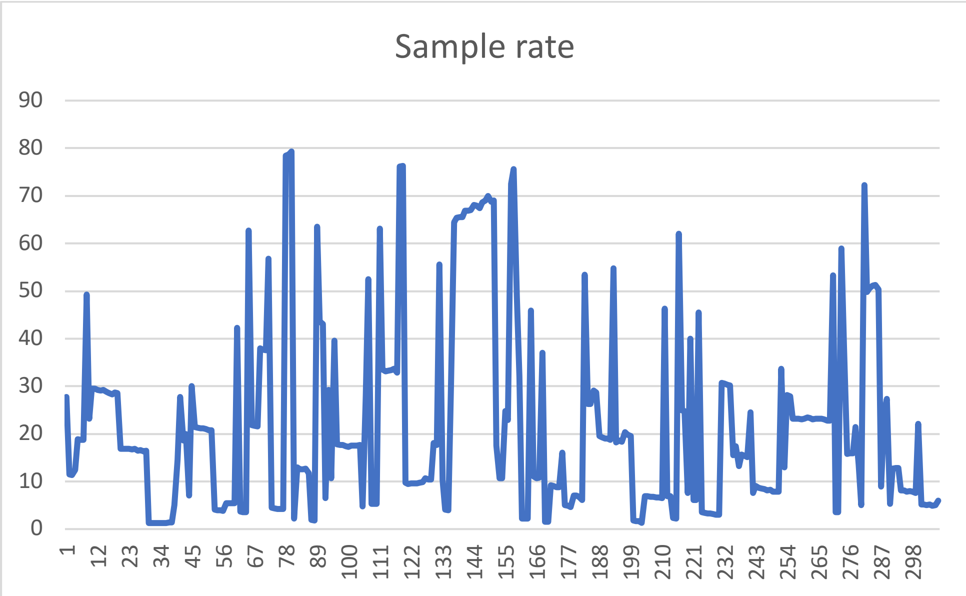


Figure D2-2: RL- based sequential sampling agent's behavior for bad accepted rate (Test 3)

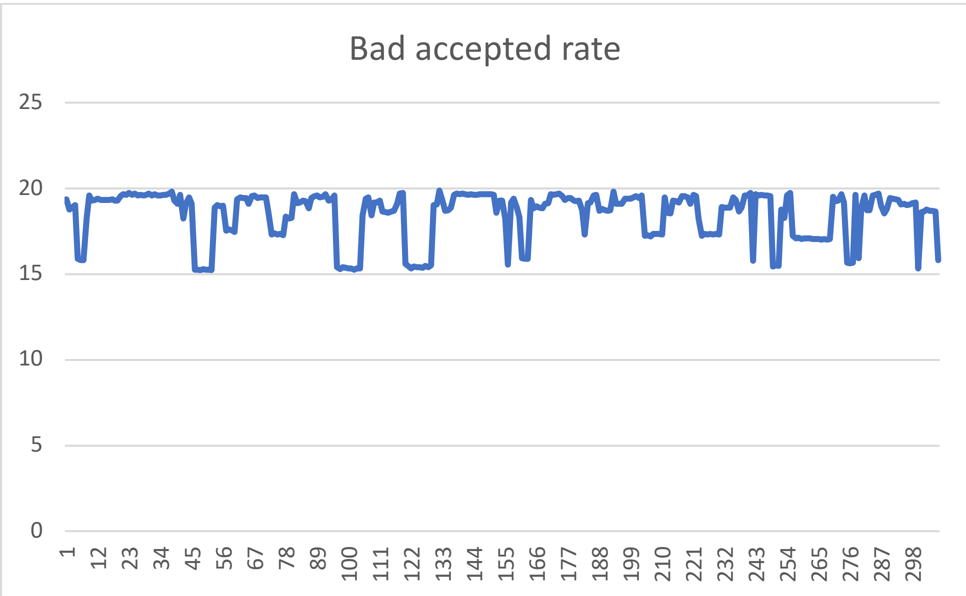


Figure D2-3: RL- based sequential sampling agent's behavior for accepted weight (Test 3)

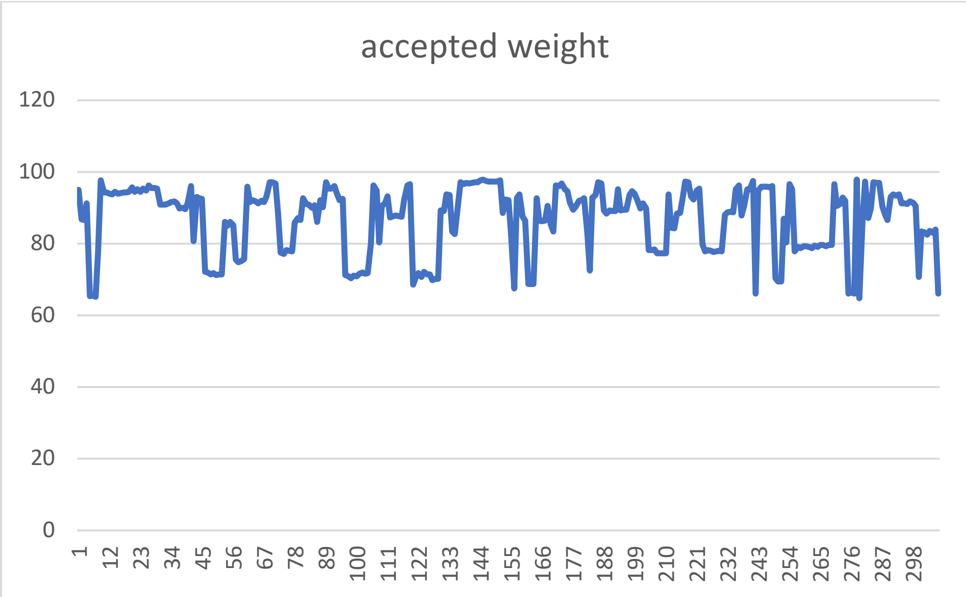


Figure D2-4: RL- based sequential sampling agent's behavior for bad rejected rate (Test 3)

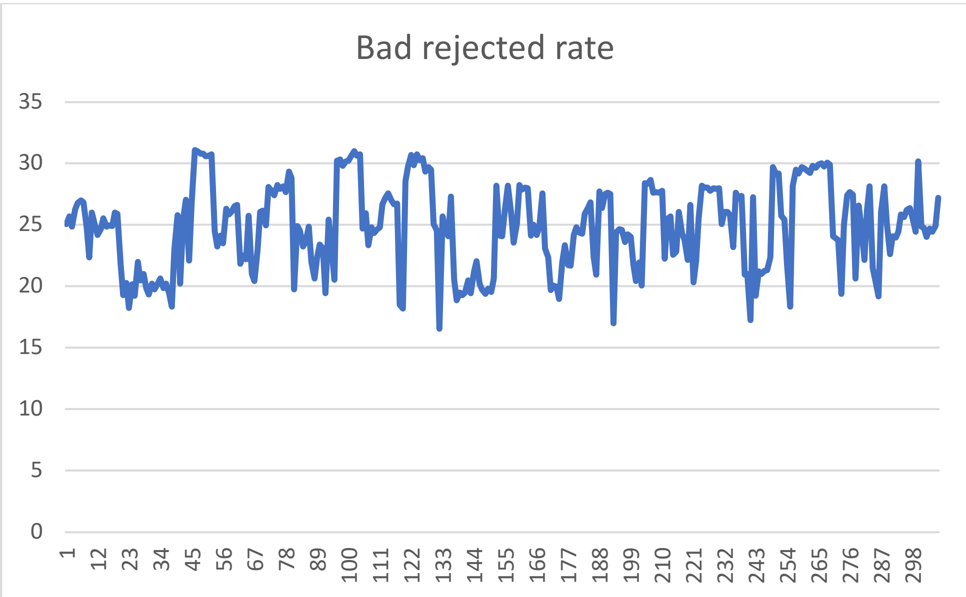


Figure D2-5: RL- based sequential sampling agent's behavior for rejected weight (Test 3)

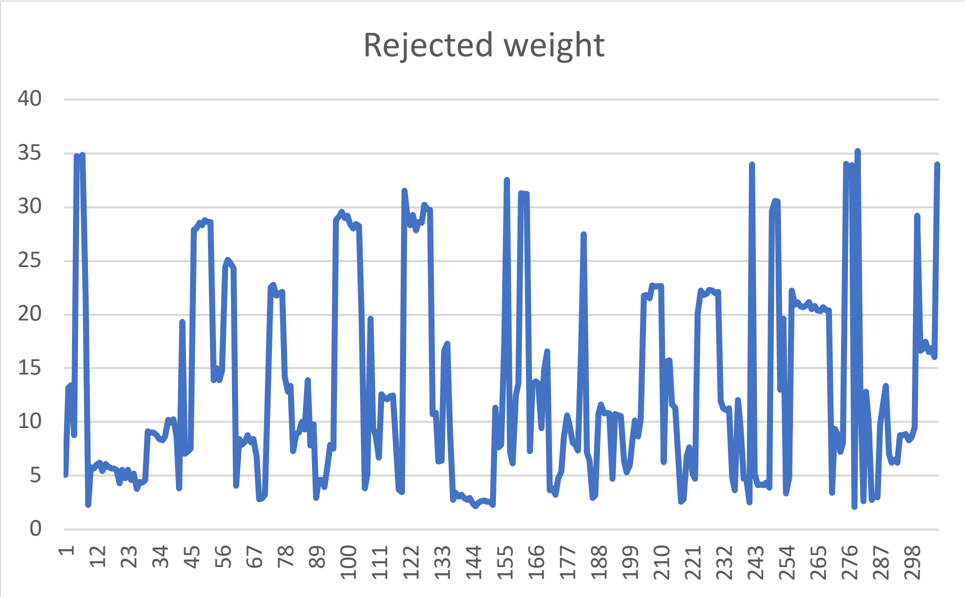
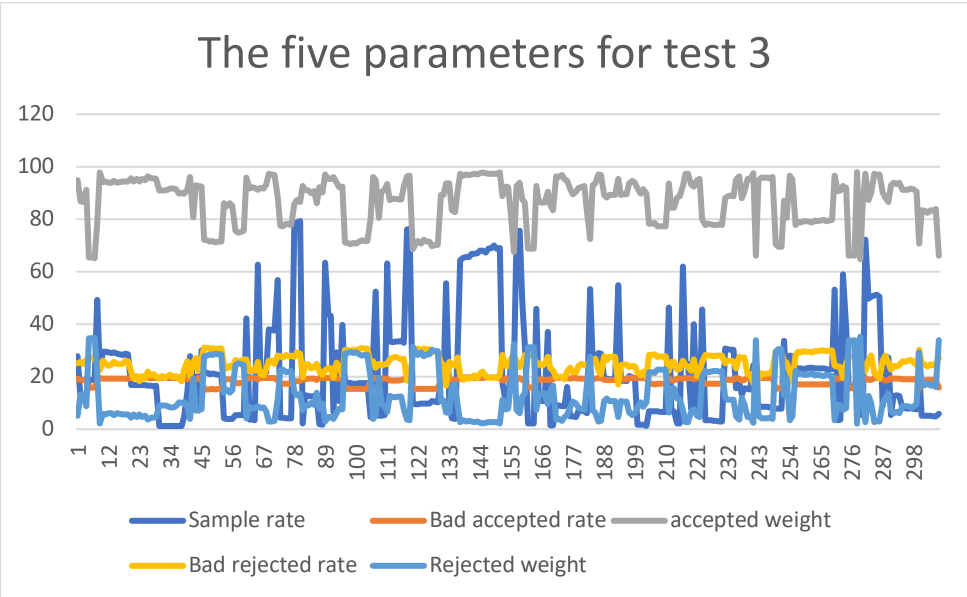
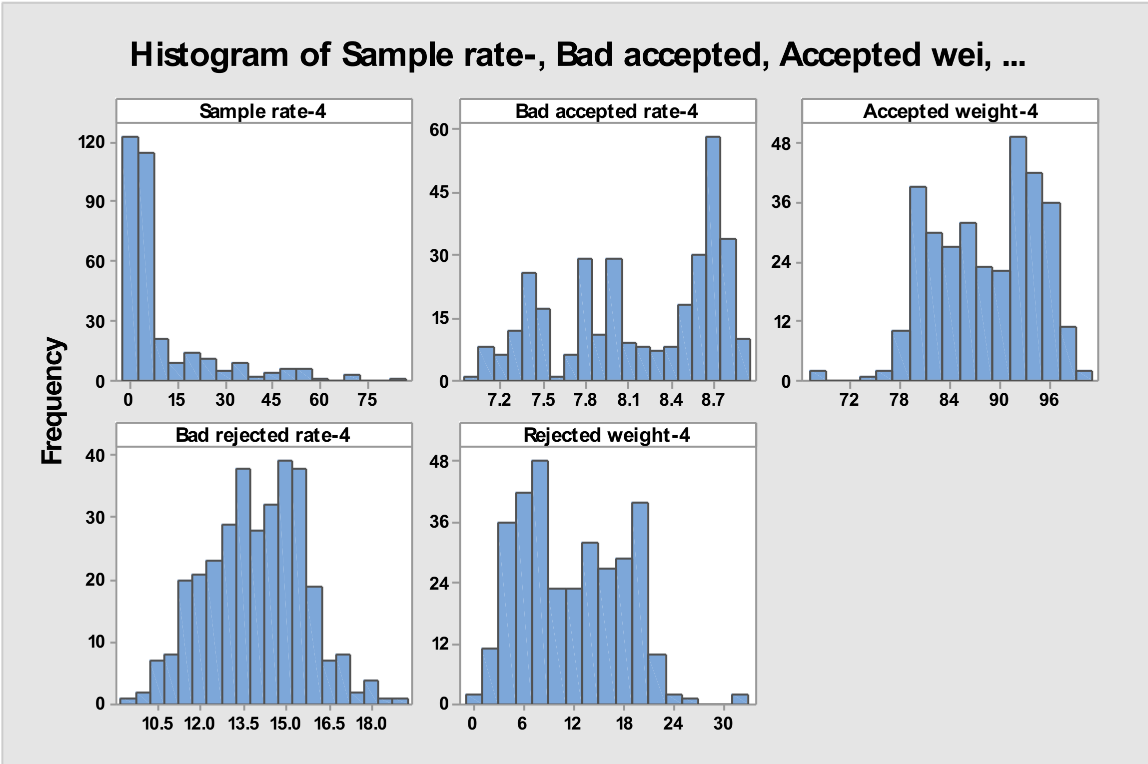


Figure D2-6: RL- based sequential sampling agent's behavior for all parameters (Test 3)



APPENDIX E: RL-Based Sequential Sampling Simulation Histogram for Lambda = 0.9 (Test 4)

APPENDIX E: RL-Based sequential sampling Histogram for Lambda = 0.9



APPENDIX E2: RL-Based Sequential Sampling Figures For Sample Size: Lambda=0.9 (Test)

Figure E2-1: RL- based sequential sampling agent's behavior for sample rate (Test 4)

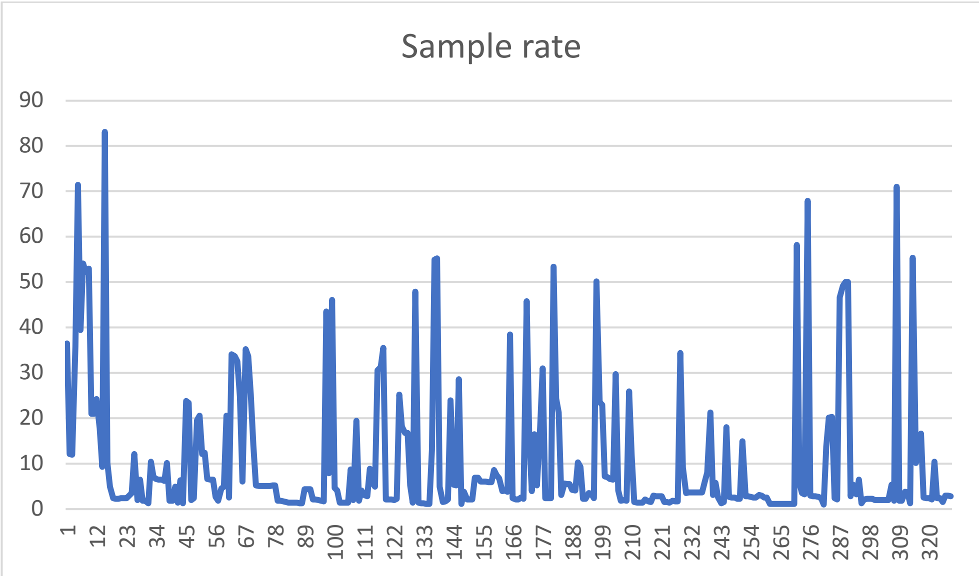


Figure E2-2: RL-Based sequential sampling for bad accepted rate (Test 4)

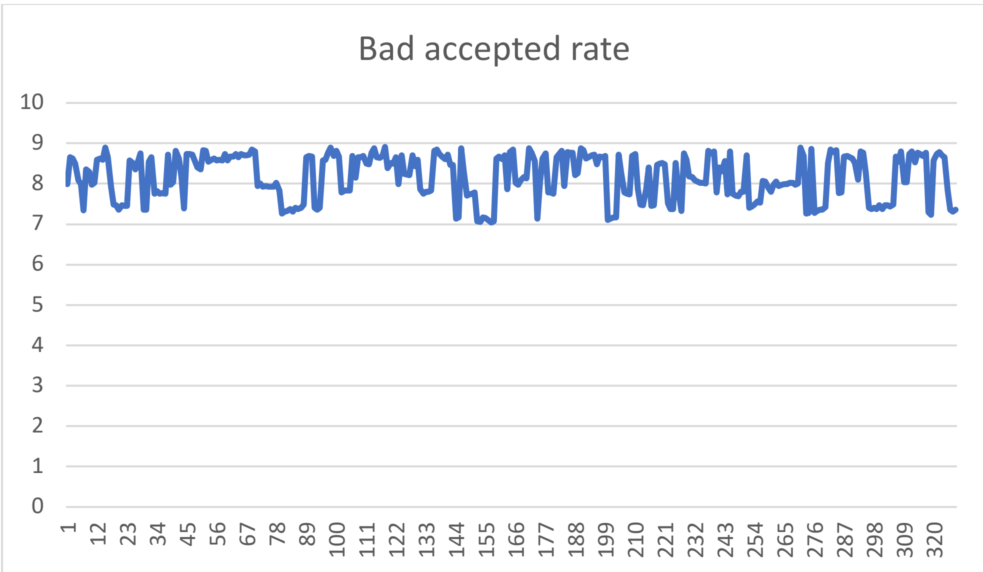


Figure E2-3: RL- based sequential sampling agent’s behavior for accepted weight (Test 4)

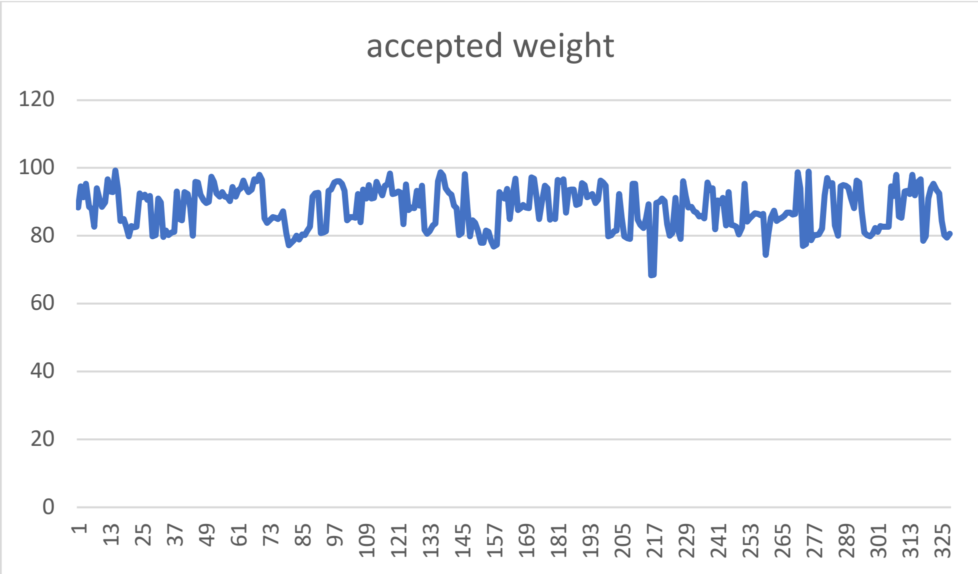


Figure E2-4: RL- based sequential sampling agent’s behavior for bad rejected rate (Test 4)

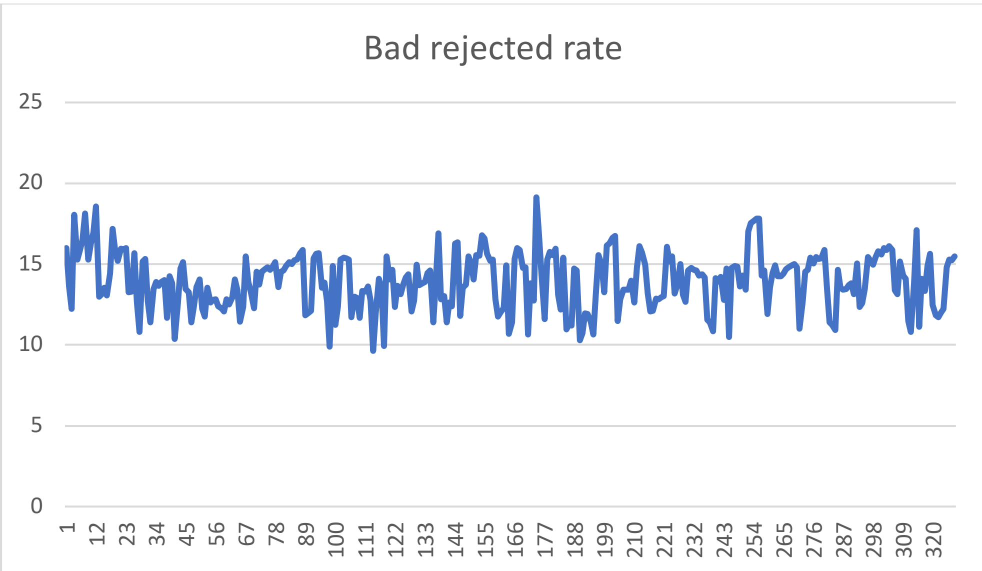


Figure E2-5: RL- based sequential sampling agent's behavior for rejected weight (Test 4)

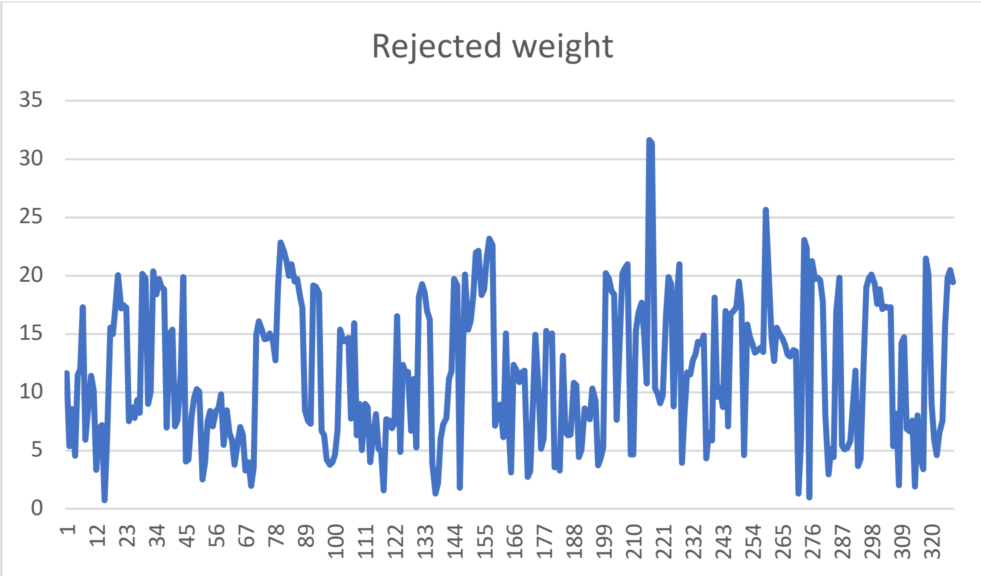
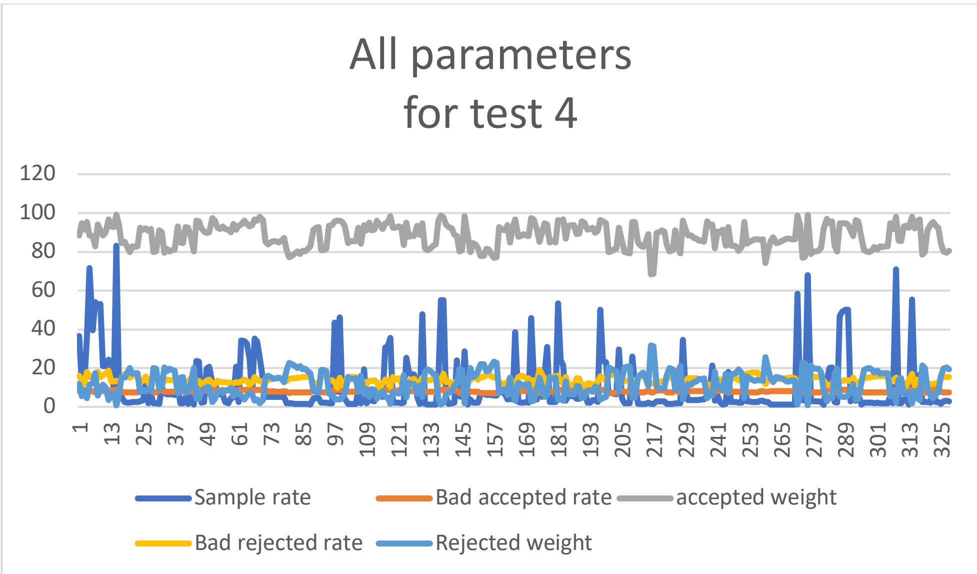


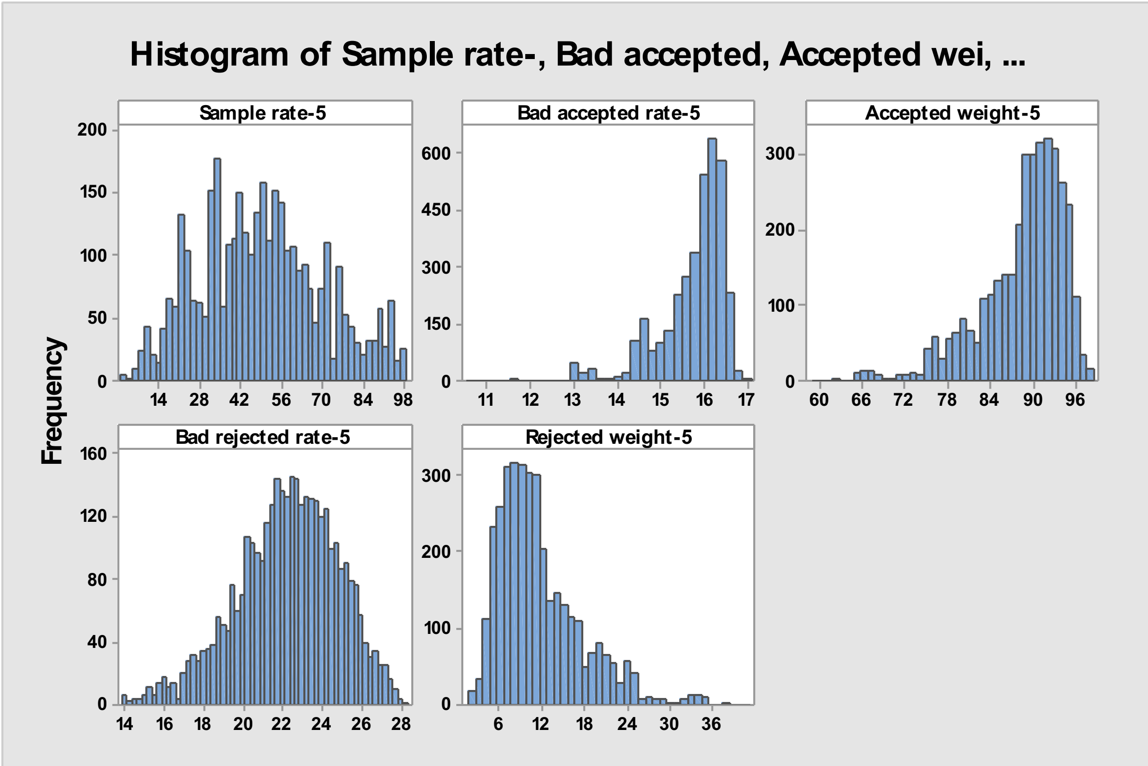
Figure E2-6: RL- based sequential sampling agent's behavior for all parameters (Test 4)



APPENDIX F: RL-Based Sequential Sampling Simulation Histogram for Beta= 5,

Lambda = 0.05, Reward (Test 5)

APPENDIX F: RL-Based sequential sampling Histogram for Beta= 5, Lambda = 0.05, Reward



APPENDIX F2: RL-Based Sequential Sampling Figures for Beta= 5, Lambda = 0.05, Reward (Test 5)

Figure F2-1: RL- based sequential sampling agent's behavior for sample rate (Test 5)

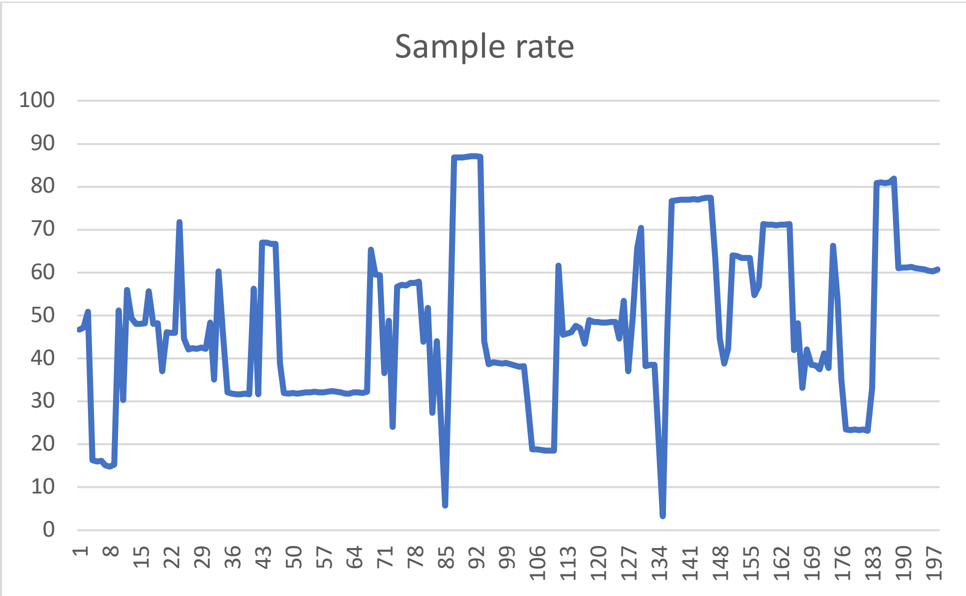


Figure F2-2: RL- based sequential sampling agent's behavior for bad accepted rate (Test 5)

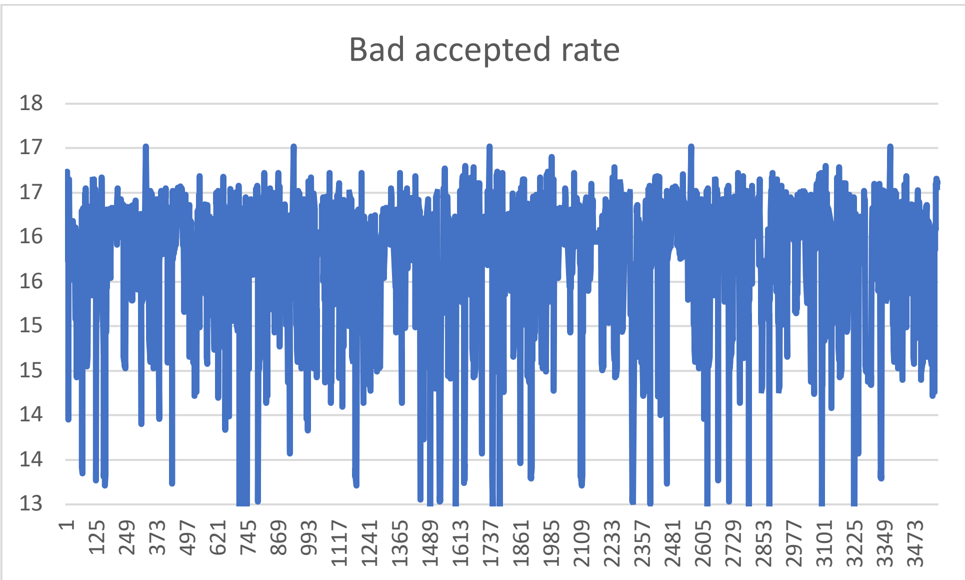


Figure F2-3: RL- based sequential sampling agent’s behavior for accepted weight (Test 5)

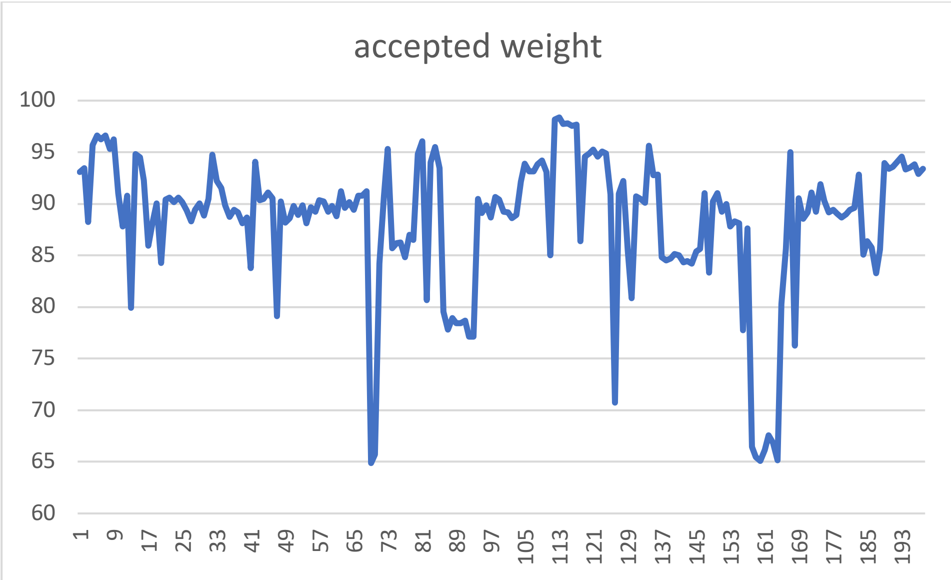


Figure F2-4: RL- based sequential sampling agent’s behavior for bad rejected rate (Test 5)

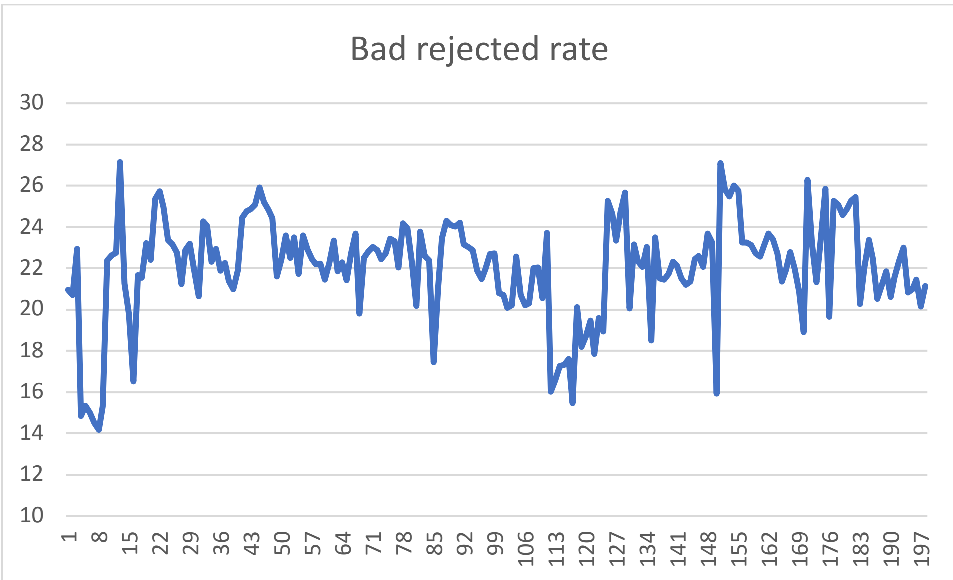


Figure F2-5: RL- based sequential sampling agent's behavior for rejected weight (Test 5)

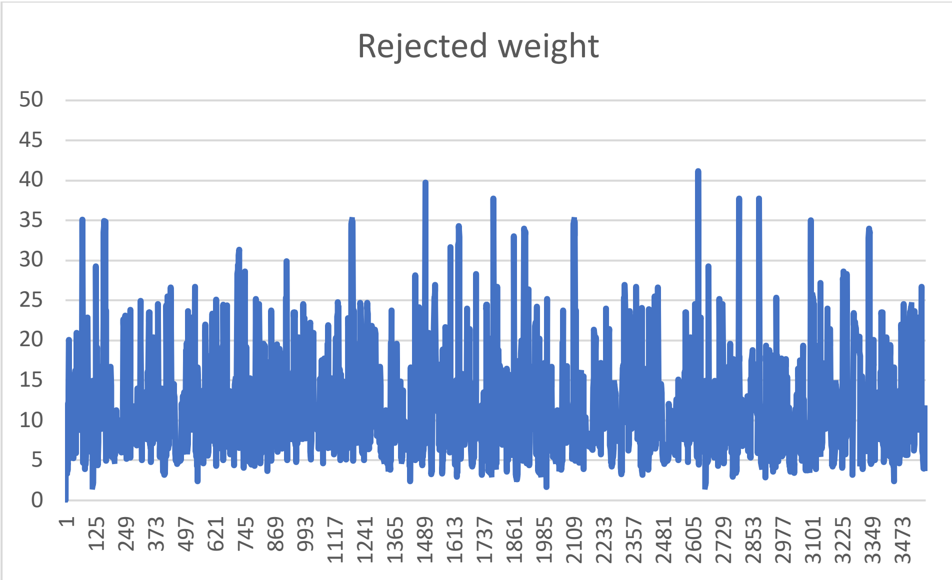
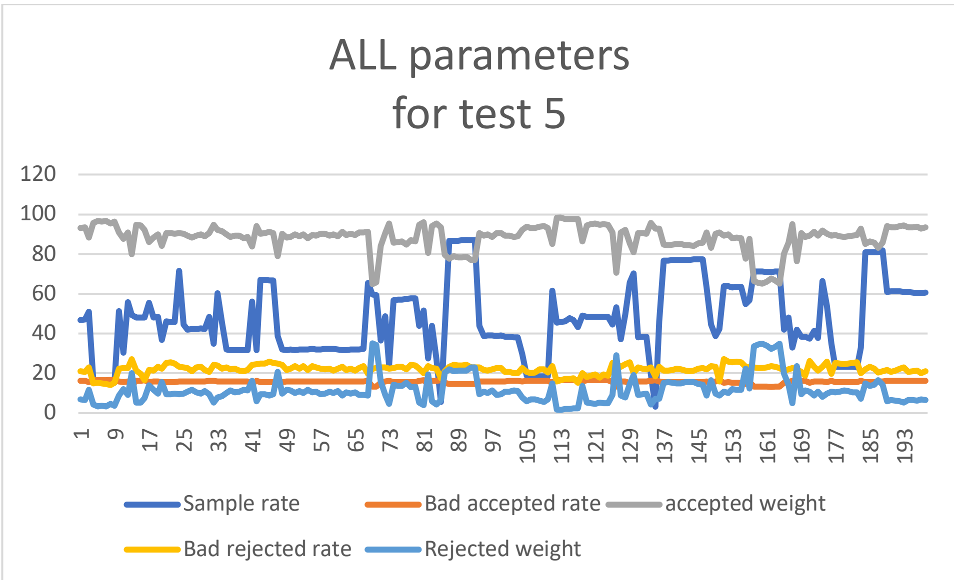
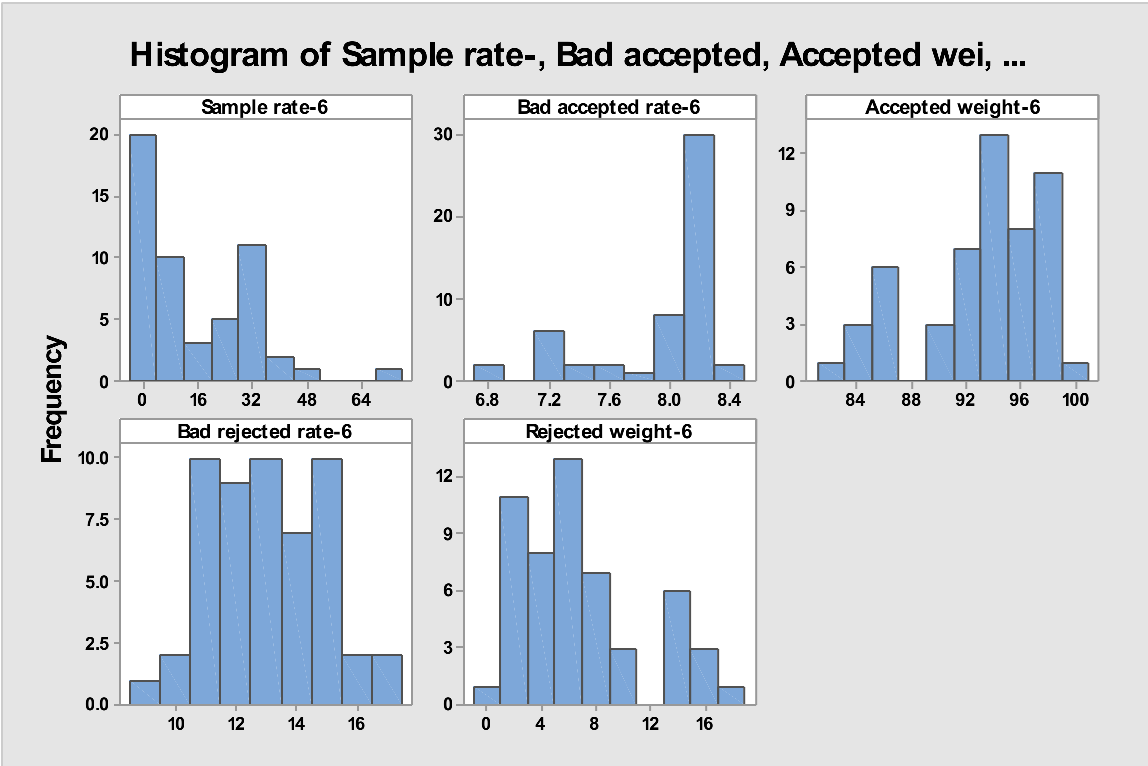


Figure F2-6: RL- based sequential sampling agent's behavior for all parameters (Test 5)



APPENDIX G: RL-Based Sequential Sampling Simulation Histogram for Lambda = 0.2, Reward (Test 6)

APPENDIX G: RL-Based sequential sampling Histogram for Lambda = 0.2, Reward



APPENDIX G2: RL-Based Sequential Sampling Figures for Lambda = 0.2, Reward (Test 6)

Figure G2-1: RL- based sequential sampling agent's behavior for sample rate (Test 6)

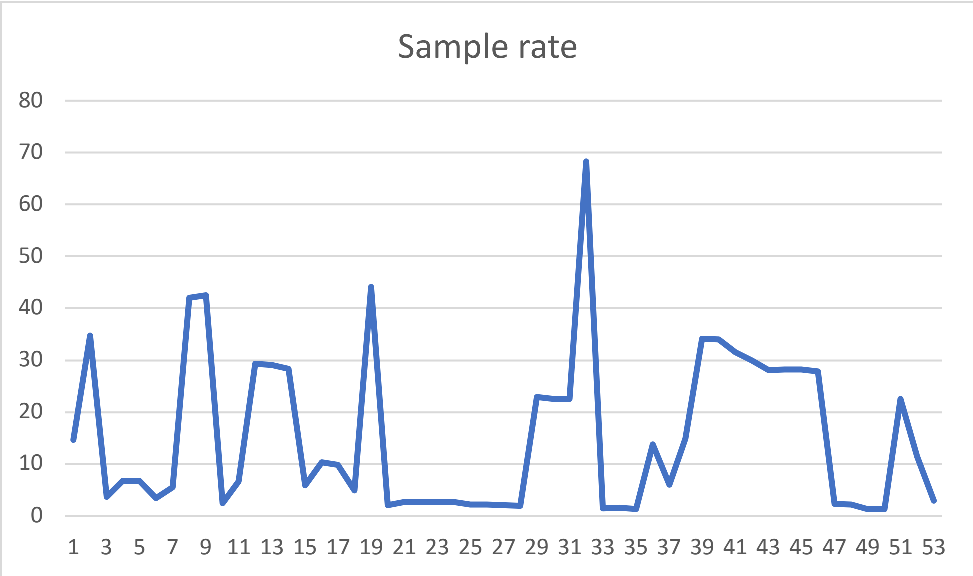


Figure G2-2: RL- based sequential sampling agent's behavior for bad accepted rate (Test 6)

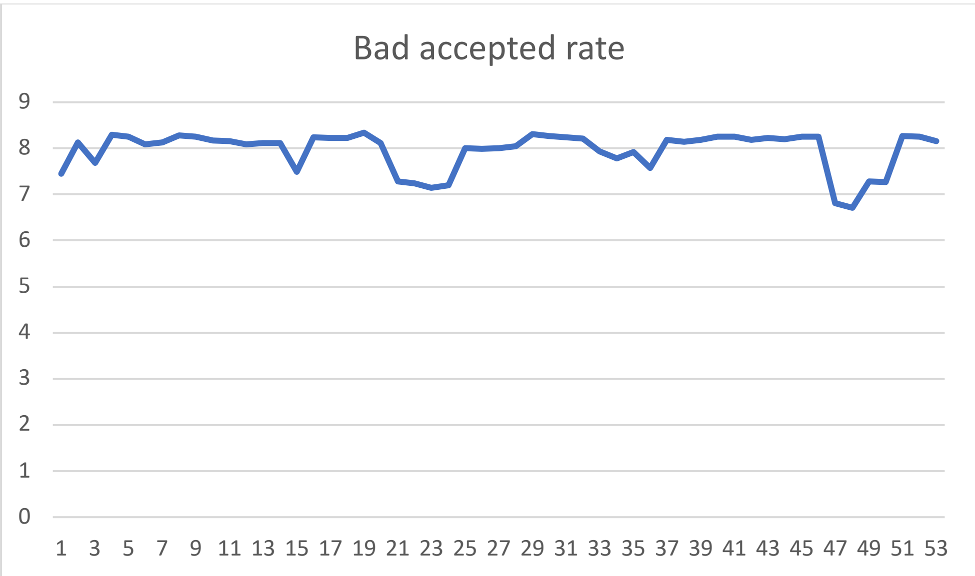


Figure G2-3: RL- based sequential sampling agent's behavior for accepted weight (Test 6)

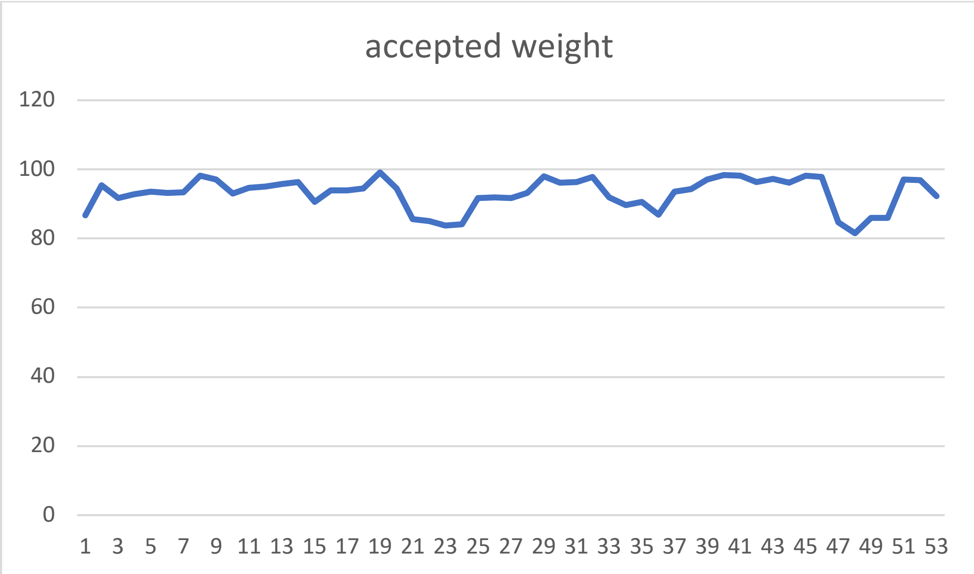


Figure G2-4: RL- based sequential sampling agent's behavior for bad rejected rate (Test 6)

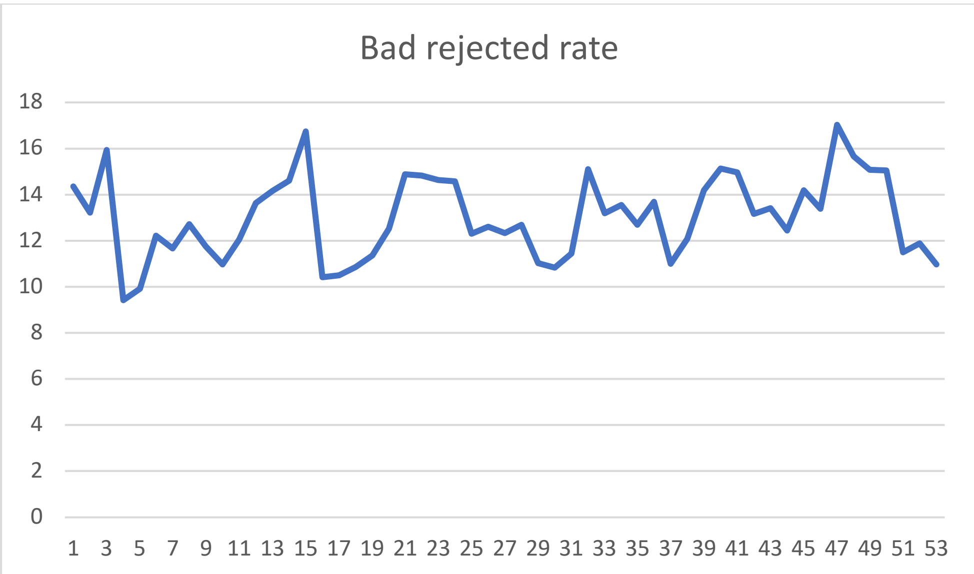


Figure G2-5: RL- based sequential sampling agent’s behavior for rejected weight (Test 6)

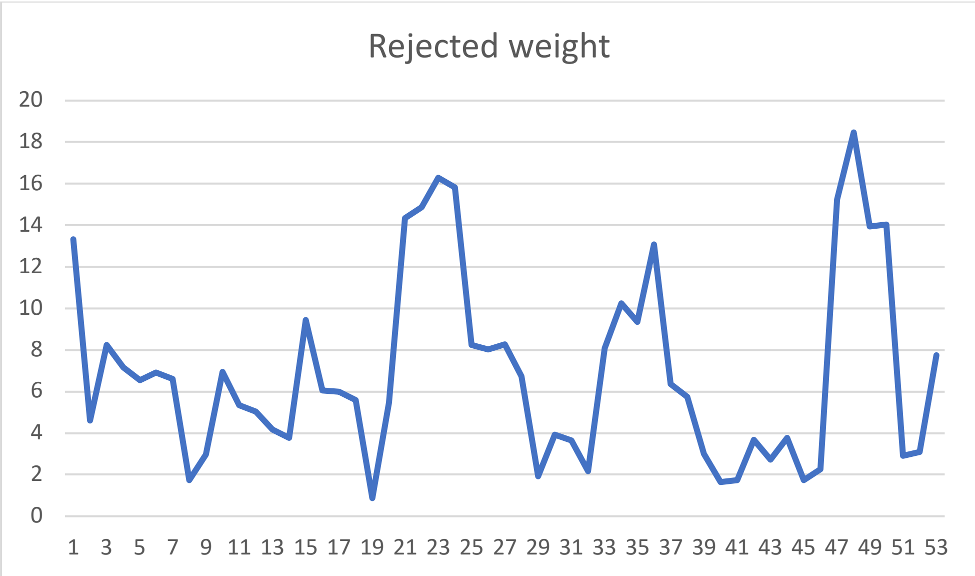
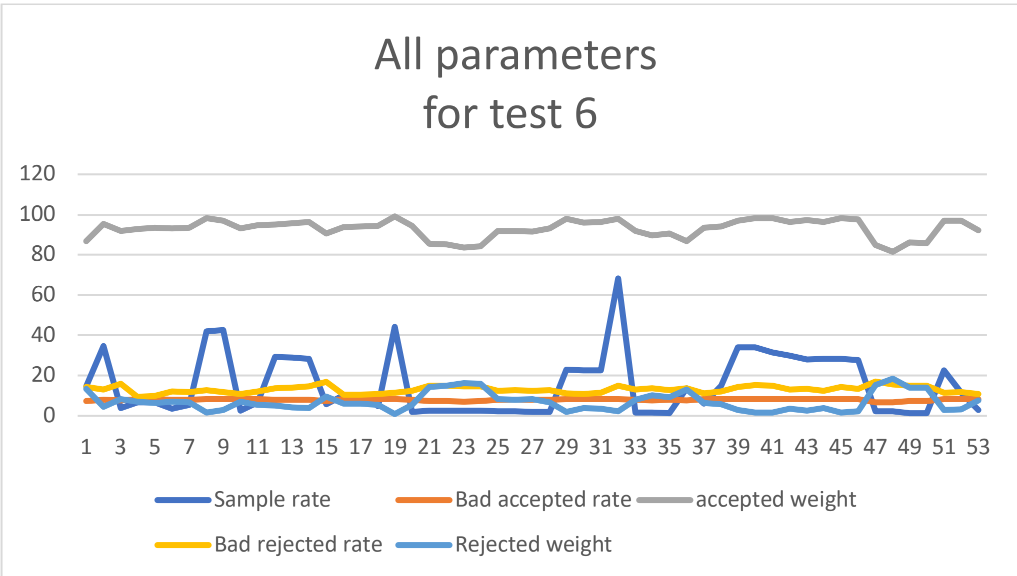


Figure G2-6: RL- based sequential sampling agent’s behavior for all parameters (Test 6)



APPENDIX H: Record of Six Simulation Runs for Each KPI (The Minimum, Maximum, Average and Standard Deviation)

Test	Evaluation criteria	Sample rate				Bad accepted rate				Accepted weight				Bad rejected rate				Rejected weight			
		Min	Max	Average	Std dev	Min	Max	Average	Std dev	Min	Max	Average	Std dev	Min	Max	Average	Std dev	Min	Max	Avr	Std dev
RL Model	RL-based Seq. Sampling	1	73	10	13	8	10	10	1	66	99	88	7	8	19	14	2	2	36	12	7
1	State thresholds constraints	1	90	12	17	8	10	10	1	75	98	88	6	11	20	15	2	2	25	12	6
2	Sample size: 8 to 50, Batch size = 500	6	76	24	15	9	10	10	0	83	94	88	3	12	19	15	1	6	17	12	3
3	Beta = 5	21	89	21	20	19	20	19	1	88	99	88	9	25	31	25	3	12	36	12	9
4	Lambda = 0.9	1	83	10	14	7	9	8	1	68	99	89	6	9	19	14	2	1	32	11	6
5	Beta= 5, Lambda = 0.05, Reward	3	98	49	21	11	17	16	1	43	98	88	7	14	64	23	7	2	57	12	7
6	Lambda = 0.2, Reward,	1	68	21	14	7	10	9	1	82	99	90	4	9	18	14	2	1	18	10	4
7	Sequential sampling	2	48	10	11	4	11	8	1	56	88	71	5	8	34	13	2	4	44	29	4
8	MIL-STD-1916 (32)	13	16	15	1	3	4	4	1	49	62	57	5	16	19	17	1	38	51	43	5
9	MIL-STD-105E (Single sampling)	9	10	10	1	3	6	5	1	59	71	64	3	12	20	18	2	10	41	36	4
10	MIL-STD-105E (Double sampling)	2	14	11	3	3	5	4	2	48	75	59	6	11	18	15	2	8	55	41	5
11	MIL-STD-105E (Multiple sampling)	3	9	10	2	3	9	5	1	46	61	51	3	10	17	16	1	13	51	49	5

Curriculum Vita

HANI ABDULWAHAB A. KHALIL
Department of Industrial and Manufacturing Engineering
University of Wisconsin-Milwaukee
3200 North Caramer Street, Milwaukee, WI 53211

EDUCATION

- **University of Wisconsin Milwaukee (UWM)** May 2020
  Degree: PhD. in Industrial and Manufacturing Engineering
  Minor: Business Administration and Supply Chain Management
  GPA: 3.81/4
  Dissertation: A Reinforcement Learning Approach to Sequential Acceptance Sampling as A Critical Success Factor for Lean Six Sigma
  Committee: Dr. Wilistar Otieno, Dr. Hamid Seifoddini, Dr. Matthew Petering Dr. Rohit J. Kate, Dr. Xiaohang Yue

- **Milwaukee School of Engineering (MSOE)** May 2012
  Degree: M.Sc. in Engineering Management
  Thesis: Implementing Lean Six Sigma Quality at MSOE's Admission Department
  GPA.: 3.61 out of 4

- **King Abdul-Aziz University. Jeddah, KSA** June 2008
  College of Environmental Design
  Degree: B.Sc. in Urban and Regional Planning
  Project: Design Complete Housing Facility for Pilgrims in Mecca
  GPA.: 3.83 out of 5

POSTGRADUATE CERTIFICATES

- **M.Sc. in Project Management** February 2012
  Milwaukee School of Engineering- The Rader School of Business

- **Lean Six Sigma Black Belt** April 2012
  Milwaukee School of Engineering- Business Excellence Consortium (BEC)

GRADUATION PROJECTS

- **A Reinforcement Learning Approach to Sequential Acceptance Sampling as A Critical Success Factor for Lean Six Sigma**
  The overarching goal of this research is to examine and develop a model based on coupling reinforcement learning methodology (RL) and sequential acceptance sampling in manufacturing to improve and achieve the optimal sample size and product monitoring.

- **Implementing Lean Six Sigma Quality at MSOE's Admission Department**
  The goal of the project was to increase Saudi Students yield at MSOE from the current situation 40 % to 70%, by develop process that specifically reach causes of Saudi students' attrition.

- **Design Complete Housing Facility for Pilgrims in Mecca**
  The project was to define the social and economic problems in Mecca as well as design a housing facility in area in a special location with easy access to all holy areas, and with complete services to pilgrims.

PRACTICAL EXPERIENCE

- **Graduate Teaching Assistant** January 2015 – December 2019
  University of Wisconsin-Milwaukee

- Quality Control - IME 571 (Spring 2015, 2016, 2017)
- Engineering Economic Analysis - IME 368 (Fall 2015, 2017 / Spring 2018, 2019)
- Methods Engineering - IME 470 (Fall 2018, 2019)
- Lean Manufacturing - IME 587 (Spring 2015)

- **Quality Assurance Assistant Manager** June 2012 – April 2013
  KAFAK International Co., Jeddah, KSA

- Gathered and analyzed information about competitors
- Implemented Lean Six Sigma methods
- Maintained the work quality

- **Urban Planner** February 2003- June 2008
  Al Tasan Consultancy Bureau, Jeddah, KSA

- Topographic survey studies of King Abdullah Economic City
- Site supervising
- Prepared and evaluated reports

- **Assistant Urban Planner**. June 2007- August 2007
  Thra'a Alkhair General Contracting Est. Jeddah, KSA

- Designing and Drawings on AutoCAD
- Prepared reports
- Followed up on government documents

AREAS OF INTEREST

Quality Control, Quality Assurance, Lean Manufacturing, Managerial Decision Making, Operations and System Management, and Machine Learning.

AWARDS AND HONORS

- King Abdullah scholarship award-Saudi Arabia (2014 - 2018)
- University of Wisconsin-Milwaukee Chancellor's Graduate Award, Department of Industrial Engineering (2018)
- INFORMS Professional Colloquium (IPC) Award (2018, and 2019)
- University of Wisconsin-Milwaukee Travel Fellowship Award (2017 - 2019)

TECHNICAL SKILLS

- Excellent in analyzing information by Minitab, SPSS, and Python
- Skilled in machine learning (Reinforcement Learning)
- Skilled at Microsoft Office applications (Microsoft Project, Visio, Excel, etc.)
- Skilled in Lean Six Sigma tools

PROFESSIONAL ACTIVITIES AND SERVICES

- **Membership**

- INFORMS
- INFORMS Professional Colloquium (IPC)
- IEEE

- **Services**

- INFORMS - UWM Student Chapter: President (2018-2019)
- Session Chair: Manufacturing II, 2019 INFORMS Annual Meeting, Seattle, WA.