THE RELATIONSHIP BETWEEN COMPONENT AND PRODUCT QUALITY IN MANUFACTURING, WITH EMPHASIS ON COMPETITIVENESS

by

Yue Wang

A Dissertation

Submitted to the Faculty of Purdue University
In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

School of Environmental and Ecological Engineering
West Lafayette, Indiana
May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. John W. Sutherland, Chair
School of Environmental and Ecological Engineering

Dr. Fu Zhao
School of Mechanical Engineering
School of Environmental and Ecological Engineering

Dr. J. Stuart Bolton
School of Mechanical Engineering

Dr. Roshanak Nateghi
School of Industrial Engineering

Approved by:
Dr. John A. Howarter

To my parents and my homeland

献给我的爸爸妈妈和我的故乡

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. John W. Sutherland, for his continuous support, guidance, and encouragement in the past five years. He is always a source of wisdom and a drive for excellence. I would also like to thank my committee members Dr. Fu Zhao, Dr. J. Stuart Bolton, and Dr. Roshanak Nateghi for their support and advice. I am grateful to my advisor during my graduate study in China, Dr. Chen-Han Lee, who first taught me how research works.

This work was supported, in part, by IN-Mac, Discovery Park Big Idea Challenge Grant at Purdue University, and Cummins Inc. Many people at Cummins Inc. have supported me, special thanks to Mike Perry, Dane Whitlock, and Girish Kumbhar.

My colleagues and friends at EEE have made my study and life at West Lafayette enjoyable. I would like to particularly thank Matt Triebe, Wo Jae Lee, Nehika Mathur, Enze Jin, Hongyue Jin, Justin Richter, Gamini Mendis, Haiyue Wu, Debora Maia Silva, Carolyn Quigley, and Aihua Huang. I acknowledge the support from staffs at EEE, especially Steve Shade, Patti Finney, Jill Wable, Cresta Cates, and Dr. Nina Robinson, who made EEE a nice place to study and work. I thank visiting scholars Lirong Zhou, Tianhua Zhang, Lei Li, Wen Tong, and Shitong Peng for enlightening discussions and collaborations. I am thankful to my friends Ying Song and Xinghe Yang for their encouragement through many text messages and phone calls. I would like to thank my teacher Xudong Jin, who believed in me more than I did.

Finally, I thank my family for their love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The capability of continuously producing good quality products with high productivity and low cost is critical for manufacturers. Generally, products are made up of components, which enable the product to perform its purpose. A complex product may be assembled from many components through multiple assembly stages. Any quality defects in a component may build up in the product. A good understanding of how the quality of components impacts the quality of products in a complex manufacturing system is essential for keeping the competitiveness of a manufacturer.

In this research, a series of quality management models are proposed based on studying the relationship between component quality and product quality. Optimal quality control leads to increased competitiveness of a manufacturer, since it helps reduce cost, increase production, and limit environmental impact. The research starts from studying the tolerance allocation problem, which is fundamental of managing the tradeoff between quality, productivity, cost, and waste. First, a tolerance allocation method that minimizes cost is proposed. This model jointly considers process variation and tolerance specifications. The relation between manufacturer, user, design, and processing are embedded in the cost model. To solve the tolerance allocation problem from the root cause, i.e., the variations in production processes, a second tolerance allocation model is then provided. This model considers both product design (tolerance selection) and operation planning (or production rate selection). Relations among production rate, production cost, processing precision, and waste are considered. Furthermore, a new process control model that extends traditional statistical process control techniques is proposed. Data acquired from a manufacturing system are usually in the forms of time series, and anomalies in the time series are generally related to quality defects. A new method that can detect anomalies in time series data

with long length and high dimensionality is developed. This model is based on recurrent neural networks, and the parameters of the neural networks can be trained using data acquired during routine operation of a manufacturing system. This is very beneficial because often, there are few data labeled as anomalies, since anomalies are hopefully rare events in a well-managed system. Last, quality control of remanufacturing is studied. A component-oriented reassembly model is proposed to manage the varied quality of returned component and varied needs of customers. In this model, returned components are inspected and assigned scores according to their quality/function, and categorized in a reassembly inventory. Based on the reassembly inventory, components are paired under the control of a reassembly strategy. A reassembly-score iteration algorithm is developed to identify the optimal reassembly strategy. The proposed model can reassemble products to meet a larger variety of customer needs, while simultaneously producing better remanufactured products.

In summary, this dissertation presents a series of novel quality management models to keep manufacturers' competitiveness. These models are based on studying factors that impact component and product quality at multiple stages of a product life cycle. It was found that analyzing the relationship between component and product quality is a very effective way of improving product quality, saving cost, and reducing environmental impact of manufacturing.

1. INTRODUCTION

1.1 Problem Description and Motivation

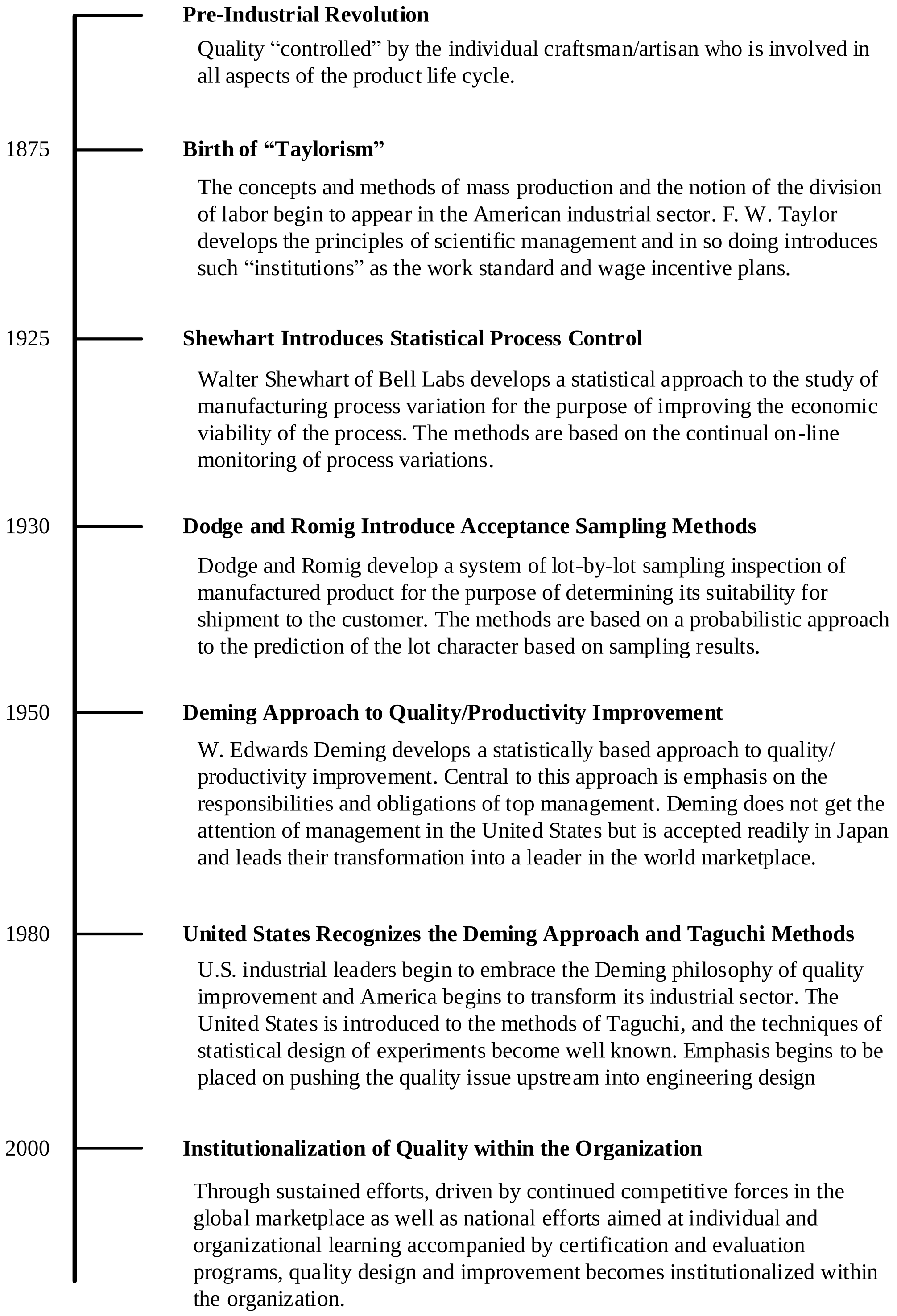
Products are systems that are produced by manufacturers and sold to customers to perform a purpose. The ability to produce high quality products with high productivity and low cost is critical for manufacturers to stay competitive. Generally, products are made up of components, which enable the product to perform its purpose. In this research, a series of quality management models are proposed based on studying the relationship between component and product quality.

As defined by Juran, quality is “fit for purpose” [1]. Functional deviation of a product from its purpose generally is associated with deviation of a component/product from its engineering specifications. And such deviations produce negative impacts on the manufacturer, and perhaps even the customer if the quality problem is not addressed by the manufacturer. For example, when variation of a process is high, the productivity will decrease since more time will be spent on inspecting and repairing defective components/products. Costs and waste will be incurred by failed components/products because of rework or recycling. Product failure causes high warranty cost and hurts customer satisfaction, which may lead to decreased market share of the products.

To reduce the negative impact of quality deviation, manufacturers usually put significant amounts of efforts into quality improvement. These efforts may have direct benefits, such as reducing defective products that are delivered to customers. Furthermore, the pursuit of high-quality helps builds the idea of never-ending improvement, which also benefits a manufacturer in the long run. For example, statistical process control is one of such techniques. It is used to increase the consistency of product/process performance. With an increased process performance, less resources would be spent on corrective actions that do not bring direct value, such as inspecting, repairing, and recycling. These strategies help continually increase the competitiveness of the manufacturer, with an increased productivity, increased market share, and reduced cost.

Increasing the quality of products has been an everlasting goal in the manufacturing industry. Quality control has been applied before the industrial revolution, and it has been improved along with the development of manufacturing technologies and management methods. And in the past decades, with the globalization, the capability of effective quality management has been critical for the survival of manufacturers.

A historical perspective of quality control is summarized by DeVore et al [2]), as given in Figure 1.1. Before the industrial revolution, individuals such as craftsman are responsible for managing product qualities. In the nineteenth century, when the industry widely adopted mass production, laborers were divided to sections of assembly lines and were only responsible for part of the production. The lack of involvement in the whole production made the laborers acquires less ownership of the products, which brought about the reduction of product quality and productivity. Industrial management pioneers such as Frederick Winslow Taylor proposed scientific management in manufacturing, which standardized the evaluation of workers' performance, i.e., the number of units produced per unit time. These efforts solved the productivity problem, but the quality issues remained. Walter Shewhart found the relation between a steady manufacturing process and the quality of products. Statistical approaches were then introduced to detect variations in a process. Causes that incurred the variations are generally the sources of quality defect. Detecting and then removing them may help solve quality issues timely. At the beginning of the twentieth century, quality control was considered a separate task from production. Quality assurance was usually carried out by a team separate from the production team. The production team is responsible for increasing the production, and the quality team was responsible for inspecting all the products and detecting the unsatisfactory ones (products that were outside of tolerances). This approach generally has low efficiency and may not be suitable for mass productions. To solve this issue, H.F. Dodge and H.G. Romig proposed the “acceptance sampling” methods, which inspect quality of small samples as a judgement of the quality of lots of goods. Starting from the latter half of the twentieth century, global competitiveness became an important issue for manufacturers. Pioneers such as Deming and Taguchi proposed statistically based quality management techniques. These techniques emphasized the importance of design on quality control. Product characteristics were designed in such a way that they are not sensitive to the change of environment. Design of experiments were widely applied in this process. Because design generally has a big impact on product quality but a low cost, the design-based approach greatly helped manufacturers increase product quality and reduce cost. In the past decades, manufacturers have obtained a better understanding of how other stages of production, such as packing, delivery, and field service, impact product quality [1]. Quality management is now generally considered at an institutional level, i.e., it is involved in multiple departments, such as design, supply chain, processing, inspection, and after market.



Pre-Industrial Revolution

Quality “controlled” by the individual craftsman/artisan who is involved in all aspects of the product life cycle.

1875 **Birth of “Taylorism”**

The concepts and methods of mass production and the notion of the division of labor begin to appear in the American industrial sector. F. W. Taylor develops the principles of scientific management and in so doing introduces such “institutions” as the work standard and wage incentive plans.

1925 **Shewhart Introduces Statistical Process Control**

Walter Shewhart of Bell Labs develops a statistical approach to the study of manufacturing process variation for the purpose of improving the economic viability of the process. The methods are based on the continual on-line monitoring of process variations.

1930 **Dodge and Romig Introduce Acceptance Sampling Methods**

Dodge and Romig develop a system of lot-by-lot sampling inspection of manufactured product for the purpose of determining its suitability for shipment to the customer. The methods are based on a probabilistic approach to the prediction of the lot character based on sampling results.

1950 **Deming Approach to Quality/Productivity Improvement**

W. Edwards Deming develops a statistically based approach to quality/ productivity improvement. Central to this approach is emphasis on the responsibilities and obligations of top management. Deming does not get the attention of management in the United States but is accepted readily in Japan and leads their transformation into a leader in the world marketplace.

1980 **United States Recognizes the Deming Approach and Taguchi Methods**

U.S. industrial leaders begin to embrace the Deming philosophy of quality improvement and America begins to transform its industrial sector. The United States is introduced to the methods of Taguchi, and the techniques of statistical design of experiments become well known. Emphasis begins to be placed on pushing the quality issue upstream into engineering design

2000 **Institutionalization of Quality within the Organization**

Through sustained efforts, driven by continued competitive forces in the global marketplace as well as national efforts aimed at individual and organizational learning accompanied by certification and evaluation programs, quality design and improvement becomes institutionalized within the organization.

Figure 1.1. Historical perspective of quality control (adapted from [2]).

Complexity of products has been continuously growing, as well as the manufacturing systems that produce the products. This brings new challenges to quality control. As variation in manufacturing is unavoidable, products cannot exactly meet the designed nominal values. In mass production, tolerances are assigned to components to guarantee quality and interchangeability (beneficial for increasing productivity and saving cost). Product designers determine characteristics of products based on their functionality (purpose of product). Product designers assign tolerances to the quality characteristics of the assembly. Process designers then “translate” the tolerances to process parameters. A complex product may be assembled at multiple stages. Variations at these stages will build up into the final products. How to optimize the tolerances of components considering the variations across multiple components through the assembly stages is very difficult and important. If tolerances are too loose, components with larger variations will be assembled, these variations will stack-up in the products, thus cause quality deviation of product. On the other hand, if tolerances are too tight, unnecessary scrap/rework of components are incurred, and more precise equipment may be deployed, which increase costs.

If anomalies of a process can be identified, it may reveal sources of quality deterioration, such as wear of cutters. Many anomaly detection techniques have been proposed, these techniques are extensions of the traditional statistical process control. The advent of low-cost sensors, wireless communication, and advances in computing, which collectively are often conveyed as the industrial internet of things, has made available large amounts of data to manufacturers. This data, from multiple sensors, is collected over time and forms a multidimensional time series. The accumulation of large amounts of time series data brings opportunities to better monitor the manufacturing system, but the anomaly detection process may also become more challenging, because the sensor data may be of longer length and higher dimensionality.

With increased awareness of the environmental issues attributable to manufacturing processes and products, researchers have started to investigate methods to make manufacturing processes and products more sustainable. Extending a product into multiple life cycles is one of these techniques. This has presented a new challenge, i.e., how to manage product quality across multiple life cycles. Once a product reaches end-of-life, it can be returned to manufacturers and may be repaired and remanufactured for another life cycle. One difficulty in these procedures is controlling the quality of the returned products/component, also known as the cores. After one life cycle, the quality levels of the cores may be very different. How to manage the cores so that they can be returned into the market with a satisfactory cost and quality is an important research topic.

The challenges summarized above show that quality design and validation affect many stages of a product life cycle. Figure 1.2 provides the relation between product quality control and the stages of a product life cycle. If some of the stages are ignored, conflicted results that causes a waste of resource may be incurred. For example, if the capability of the process is ignored in design, an unnecessary high-quality requirement may be requested. Process with low efficiency may be applied to achieve this requirement (such as low feed rate in machining), and unnecessary scrap of good components may occur. This is a waste of energy and resources. On the other hand, if the cost of processing is reduced by sacrificing quality, this may lead to products that may fail early in their life cycle. The energy and resources embedded in these products are wasted. An overall view of the whole product life cycle is necessary to improve this capability.

Few studies on quality control consider multiple life stages of a product. In the next section, a literature review is carried out. It is focused on quality control, and more specifically, on tolerance allocation, anomaly detection in production processes, and quality control in reassembly.

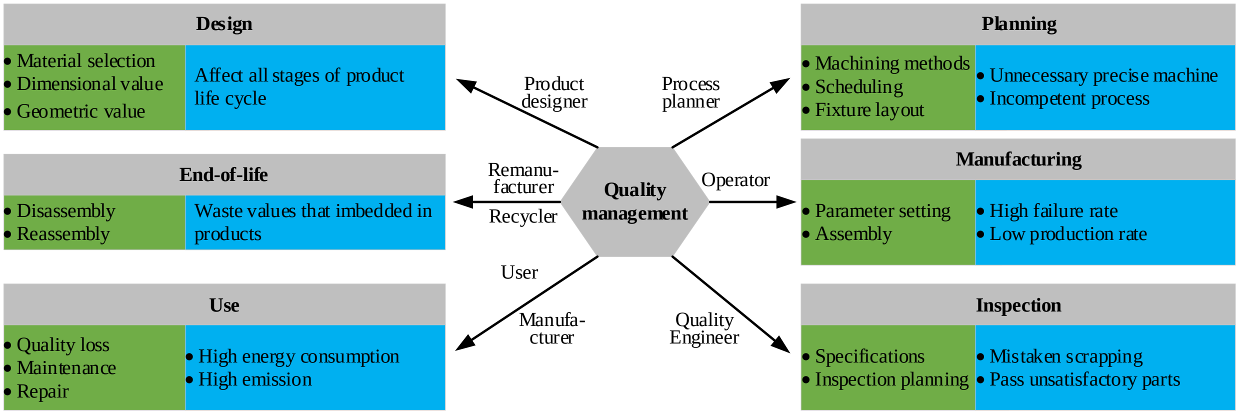


Figure 1.2. Quality management on multiple phases of a product [3].

1.2 Literature Review[1]

1.2.1 Tolerance Allocation

Most studies on tolerance allocation have focused on minimizing cost using tolerance-cost models. Various forms of functions have been proposed to model the tolerance-cost relationship [4] [5] [6] [7] [8] [9]. In some production systems, inspection is carried out on individual components. If a component fails to meet the specification, it will be scrapped, and this creates another cost. Some research has incorporated inspection and scrap costs in tolerance allocation models [10] [11]. To consider the broader cost caused by deviation of product quality characteristics from their nominal values, some studies have included quality loss in their models [12]–[14]. Various forms of functions have been proposed to model the tolerance-cost relation. Based on these tolerance-cost models, the task of optimally allocating the tolerance is usually transformed to a constrained optimization problem. The objective of this optimization problem is to minimize cost, the constraints are tolerance requirements, and the optimization variables are tolerances of individual components.

---

[1] Reprinted with permission (portions enhanced/adapted) from [3], [141]–[144].

Few studies on tolerance optimization consider the linkage between design and production. For the few studies that have considered this linkage, the impact of processing parameters on tolerance for specific machining processes are considered. Wang and Liang [15] proposed a tolerance allocation method that minimizes machining cost. Based on the relation between cost and machining parameters, the tolerance of components, the machining sequences, and machining parameters are selected. Liu and Qiu [16] integrated machining time into the tolerance optimization model. The methods that consider both design and production can help practitioners estimate how machining conditions for common machining methods affect the overall design of tolerance. Process variables, such as feed rate and cutting speed in machining or scanning speed in additive manufacturing, directly affect the cost and precision (variation) of a process [17]. A process usually has multiple variable settings (e.g., cutting speed, cutting depth, and feed rate in machining), and the impacts of each variable on the cost and process variation are different [15] [18]. The coupling effect among different conditions/variables makes it difficult to find the optimal variable settings (for a variety of objectives), and the optimization result for one type of process are difficult to generalize to other types of processes.

Another difficulty on tolerance allocation is predicting how the variations of components stack-up into the overall variation of the product. Traditionally, this task has been solved using simple approaches such as a worst-case model or root sum square model [15] [19]. Generally, simple models are only applicable to simple assemblies. For a complex assembly, tolerances optimized by simple models may result in high manufacturing costs due to excessively tight tolerances.

One of the most common ways to address the tolerance allocation problem for complex assemblies has been heuristic methods. Heuristic methods have the potential to find optimal or near-optimal solutions to complex problems. Singh et al. [20] and Haq et al. [21] used genetic algorithms to optimally allocate product tolerances. Zahara and Kao [22] combined the Nelder-Mead simplex method with a particle swarm optimization method to minimize manufacturing cost and quality loss. Zhang et al. [23] applied a particle swarm algorithm to satisfy tolerance requirements. While these heuristic algorithms have had success in tolerance allocation, a challenge with them is that they are very sensitive to tuning parameters, and unfortunately, such parameters are generally determined by trial and error. The lack of general guidance for selecting parameters limits the wide applicability of heuristic methods to industrial practitioners.

Some researchers applied optimization strategies in the tolerance allocation problem for complex assemblies. The method of Lagrange multipliers has been used to find optimal tolerances [5] [24] [25]. The Lagrange multiplier method is a strategy that transform a constrained optimization problem into an unconstrained problem. It helps find closed-form optimal tolerances. Tlija et al. [26] combined the Lagrange multiplier method with a technique that evaluates the difficulties of manufacturing a given part. This combination enables designers to estimate the difficulty and cost of manufacturing the product by simulation. Another common method is the Lambert W function, which is usually used in physics. Shin et al. [27] used the Lambert W function to find the tolerances that minimize the summation of manufacturing cost and rejection costs. Sofiana et al. [28] used a third-party software package to solve a tolerance optimization model. This model considered the impact of rework on the quality of product, and the impact of a profit-sharing policy, which may stimulate the commitment of suppliers in quality improvement.

Simulation-based strategies have also been widely used in predicting the stack-up of component variations. Qureshi et al. [29] proposed an iterative optimization procedure based on Monte Carlo simulation. Samples of given distribution is fist generated, and then the impact of component variation on the assembly is estimated by simulation. Wu et al. [30] used Monte Carlo simulation in the statistical analysis and introduced a genetic algorithm to improve the efficiency of the estimation. Hoffenson et al. [31] provided survey-based tolerance allocation method that considers the economic and environmental impact of scrapping components and products. Design of experiment and Monte Carlo simulation are deployed to predict the stack-up of individual tolerance into a product. Haghighi and Li [32] proposed a tolerance design method for additive manufacturing. This method estimates characteristics of population using a bootstrap statistical technique, which is based on simulation. Huang et al. [33] proposed a method that optimize tolerances on two stages. First, a tolerance model is built based on sampling, then a gradient-based strategy is used to optimize the model. Rosyidi et al. [34] and Rosyidi et al. [35] proposed a simulation-based method that considered a situation where the process capability of suppliers are a variable. A fuzzy quality loss function is included in the model to consider the cost related to the quality of products.

Table 1.1. Common tolerance optimization methods and limitations.

Method	Advantage	Disadvantage
Optimize process parameters	Solve the problem from the root-cause, variation of processes.	Difficult to find the optimal variable settings. Results are difficult to generalize.
Heuristic	Can find near optimal tolerances. Do not need rigorous optimization procedures.	Results are sensitive to tuning parameters, which are determined by trial and error. Difficult to generalize.
Optimization strategies	Can be used in complex assemblies. No need of tuning parameters.	Statistical analysis is needed to build such a model, and the precision of the prediction depends on the model.
Simulation-based	No need to build statistical models to predict the accumulation of tolerances. The precision of the prediction is high.	Time-consuming. It generally needs simulation of large samples and many iterations.
Method of this chapter	Have an overall consideration on quality, cost, and waste. The linkage between design and production are considered (considers both product design and operation planning.)	

1.2.2 Anomaly Detection

The advent of low-cost sensors, wireless communication, and advances in computing, which collectively are often conveyed as the industrial internet of things, has made available large amounts of data to manufacturers [36]–[38]. This data, from multiple sensors, is collected over time and forms a multidimensional time series. The accumulation of large amounts of time series data brings opportunities to better monitor the manufacturing system, but the anomaly detection process may also become more challenging, because the sensor data may be of longer length and higher dimension. For example, every time step of the time series collected from a motor may be a vector containing acceleration signals of multiple axes and vibration signals of multiple directions for a long period of time [39] [40]. Traditional anomaly detection methods may not be capable of detecting anomalies in time series data of long length and high dimensionality [41].

Many efforts have been carried out to solve the challenge of detecting anomalies from time series data, and these efforts may be classified into two groups. The first group of studies takes a strategy that preprocesses the time series data. This can be a two-step process, with the first step finding sub-sequences within a time series that have the largest differences from other subsequences, with large differences being evidence of an anomaly [42] [43]. For a time series with a long length, the series can be divided into segments, and a clustering algorithm can be employed to locate an anomaly [44]. For a time series of high dimensionality, it can first reduce the dimensionality of the data and find key signatures using, for example, multilinear principal component analysis, then use a classification model to detect anomalies [45]. These methods have gained popularity. However, a good understanding of the structure of time series data is needed to preprocess the data, which may be a difficult task for high dimensional data. One way to avoid preprocessing of data is to use supervised learning to train models that can automatically acquire features from the data [46]–[49]. But these methods need a large number of labeled anomalies (the anomalies must be identified *a priori*) to train the model; such detailed information is usually not available [41]. Sipple [50] proposed a model that can generate samples with anomalies using anomaly-free data. Random forests and neural networks were trained using the samples to detect anomalies. Zhang et al [51] introduced a convolutional neural network to analyze the correlations among time series data. A feature map was then built to represent the temporal information. Wang et al [52] transformed a one-dimensional time series data into time-frequency images using wavelet analysis. A convolutional neural network was then used to learn features in the time series data.

A second group of studies uses semi-supervised learning or unsupervised learning to solve the problems caused by the lack of labeled anomalies [41]. The basic idea of these methods is to use time series data collected under routine operation to establish a time series model (including estimates of parameters). The differences between the raw data and model predictions serves as the residuals, and the estimation procedure is focused on minimizing these residuals. There are two common ways to form the models. The first model uses a subsequence of data from past time steps to predict data in future timesteps. Malhotra et al. [53] proposed a model to use time series data of past time steps to predict data of multiple future time steps. The data at each time step is predicted multiple times (at multiple time steps), and the distribution of the predicted data are used to predict the probability of the data being abnormal. Hundman et al. [54] proposed a similar method to predict a one-dimensional time series data for each dimension. A predetermined length of time series is used to predict the signal of the next time step. The reconstruction errors are then smoothed and monitored for anomalies. Hayton et al. [55] used data collected from a jet engine to train a support vector machine model, which built a hyper plane that could separate anomalies from anomaly-free data. A more powerful reconstruction model applies the encoder-decoder structure [51], [56]–[59]. The encoder-decoder structure was first proposed for machine translation [60] [60]. A series of neural networks are used as the encoder, which summarizes information from the time series data, then another series of neural networks are used as the decoder, which takes the information summarized by the encoder to reconstruct the time series.

One limitation of the aforementioned anomaly detection methods is the accumulation of reconstruction errors. Because of their capability in retaining information of sequential data, recurrent neural networks (RNN) are one of the most commonly used approaches to create such a reconstruction model. The reconstruction process in an RNN model is carried out recurrently; thus, prediction errors of former timesteps accumulate through time. This may limit the effectiveness of using the reconstruction errors as an anomaly detector.

Another limitation of the RNN based encoder-decoder model is that it may have difficulties characterizing dynamics of high frequency in the input data. In general, there is a tradeoff between time series model complexity and the character of model reconstruction errors (residuals). A simple model may not adequately describe high frequency dynamics in the raw data, or it may well describe high frequency content, but fail to describe a slow drift in the signal. Increasing the complexity of the model allows more features of the raw data to be described, but this comes at the expense of more parameters to estimate. For common RNNs, it is not uncommon for high frequency, low amplitude components in the raw data to appear in the reconstruction errors, even though these components may not be anomalies [61].

1.2.3 Component-Oriented Reassembly

Usually, reassembly processes of remanufacturing systems are managed in a product-oriented model [62], as the quality and functional levels of a product are the easiest to determine. In a product-oriented model, when products reach end-of-life, they are returned to remanufacturers. Often, the constituent components of the product are also assumed to have reached end-of-life, and these components are considered for remanufacturing [63]. Such an approach has two drawbacks.

First, many components are likely received before they require remanufacturing. A product is returned for remanufacturing when it fails, perhaps because of failure of a single component. But, since components fail at different rates, and for different reasons, many of the remaining components will still retain a portion of their utility and could have several years of potential service remaining before reconditioning is needed. If the components in “good” and “bad” conditions are not identified and separated from one another, and all the components are remanufactured to a “like new condition,” the potential utilities of the components in a “good” condition are effectively wasted.

Second, the product-oriented model is not flexible, because it only remanufactures products to “like new” or better condition [64]. However, customers have a wide variety of needs, and some may not require “like new” performance. When customers evaluate products by matching their needs with the functional capabilities of the products, and the products perform better than needed and cost more than the customer wants to spend, then the customer may not purchase the product [65]. Thus, the product-oriented model cannot satisfy the needs of every customer. It is better to offer a variety of different products to better satisfy different customer demands.

To avoid the problems inherent in the product-oriented model, a component-oriented model can be used. In a component-oriented model, the returned components are categorized into classes according to their quality or functional levels and can be reassembled into products with functional levels different from those of new products [66]. As the quality of the returned components are considered in the reassembly process, the timing of the return of the components and the optimal time for remanufacturing the components may be matched. The returned components can be reassembled into products with predefined qualities and functions [67] [68]. Due to this flexibility, the prices of the remanufactured products can also have a larger range than new products. Thus, a wider variety of customer needs can be satisfied, which means more components will be remanufactured and sold [69]. Some companies such as Pitney Bowes [70], Ricoh [71], Airbus [67], and IBM [72] have developed remanufactured products using a business model that is different from new products.

When dealing with the component-oriented model, many uncertainties make it challenging for the remanufacturers to control the remanufacturing system. One of the uncertainties originates from the quality of the returned components, because the remanufacturers often have little control over the quantity, quality, timing of return, variety, and complexity of the returned components. Instead, they passively deal with components from returned products of different generations and types [73] [74]. Some companies deal with these uncertainties by using manual labor to manage the materials [75], which increases the cost of operation. The uncertainty of returned components requires control strategies that cost-effectively operate the remanufacturing system and provide the required quality of products [67].

Some models have been developed to characterize the varied quality of returned components and the components quality-driven and customer needs-driven models have been developed, as summarized in

Table 1.2. These models provide insight in optimizing the reassembly process, however, the models have limitations. Research on component-quality driven models mainly focuses on the techniques to deal with the variation in quality of the returned components but ignores the variety of needs of customers. Customer-needs driven models react passively to the needs of customers, therefore, using a reassembly strategy to fulfill a certain order may not be an optimal strategy at the reassembly inventory level.

Table 1.2. Representative literature addressing uncertain quality of returned components and satisfying varied customer needs.

Focus area	Author	Brief description of work	Remarks
Varied qualities of returned components	Liu et al. [76]	Quantitatively estimates reassembly precision by studying features of recycled components.	Improves the reassembly precision but ignores variety of needs of the customers.
	Shen et al. [77]	Uses an assembly Jacobian-torsor model to analyze geometric error of remanufactured components.	Focuses on geometrical variations of returned components, ignores functional variations.
	Mashhadi et al. [78]	Analyzes the relationships between quality, buy-back pricing, and profit of used electronic components based on previous product life cycle data.	Improves understanding about collecting previous life cycle data.
	Zhang et al. [79]	Categorizes returned components into categories suitable for reuse, remanufacturing, or recycling.	Provides scientific strategies to deal with returned components with different reliability levels for remanufacturers. Functional levels are ignored.
Varied customer needs	Jin et al. [80]	Proposes an assemble-to-order model to match different quality-based orders.	Improves the flexibility of the reassembly system by interchangeably assembling components with different quality levels.
	Mont et al. [81]	Provides a product-service system that leases refurbished or remanufactured prams consider customers' need.	Examines business models which lease remanufactured products.
	Sakai and Takata [65]	Presents a reconfiguration method that remanufactures components returned from different generations of products into products with varied combinations of functions.	Provides a new concept of remanufacturing products with different performance levels and functional combinations.

1.3 Research Gaps

As analyzed in the previous two subsections, tremendous progress has been made on quality control, specifically, in the areas of tolerance allocation, anomaly detection, and quality optimization in reassembly. But some research gaps still exist.

There are two major problems in the tolerance optimization methods mentioned above. First, these methods focus on minimizing cost and ignore the corresponding waste. Failure to consider waste, i.e., focusing exclusively on cost, may result in unnecessary scrapping/recycling because of excessively tight tolerances (unnecessarily rejecting components) or quality issues in assembly that result in product rejects. Second, these methods allocate tolerances using a product/component design-oriented approach, in which almost all the focus is placed upon the tolerances of the product/components, e.g., costs are modeled as a function of tolerance. Such a focus fails to capture the linkage between design and manufacturing. Also, the root cause of the problem, i.e., variations in production, is not considered in the tolerance allocation problem.

Products and manufacturing systems are becoming more complex, a complex product may be produced at multiple assembly stages. During manufacturing processes, a large amount of data is collected. Generally, these data are of high dimensionality and high length. Traditional statistical process control methods may have difficulties analyzing these data. New data analysis techniques such as machine learning and deep learning may be combined with techniques of quality control to better manage production data.

In the component-oriented reassembly model, many uncertainties make it challenging for the remanufacturers to control the remanufacturing system. One major uncertainty originates from the quality of the returned components, because the remanufacturers often have little control over the quantity, quality, timing of return, variety, and complexity of the returned components. Also, research on component-quality driven models mainly focuses on the techniques to deal with the variation in quality of the returned components but ignores the variety of needs of customers. New control methods are needed to consider these uncertainties in remanufacturing.

1.4 Objectives, Approaches, and Contributions

Good management of product quality is fundamental of keeping the competitiveness of manufacturers. Managing product quality requires finding tradeoffs among multiple relations surrounding a product, as shown in Figure 1.3. For example, the relation between manufacturers and users, cost and productivity, design and manufacturing, and precision and waste. To have an “optimal” solution of quality management, multiple questions must be answered. For example, how could manufacturers satisfy the generally high but diversified requirements of quality from users? How to have an overall consideration of design and manufacturing? As analyzed in former subsections, components are the bridges between these relations, and thus, the core of finding answers to the questions. A series of studies are carried out to get a fundamental understanding of the relationship between component and product quality.

First, a tolerance allocation model is proposed. This model minimizes cost by jointly considering process variation and tolerance specifications. The model employed a cost model that includes processing cost, scrap cost, and quality loss (impact of quality deviation in users evaluated in a monetary scale). The relation between manufacturer, user, design, and manufacturing are embedded in the cost model.

To solve the tolerance allocation problem from the root cause, i.e., the variation in production, a second tolerance allocation model was provided. This model considers both product design (tolerance selection) and operation planning (or production rate selection). Relations among production rate, production cost, processing precision, and waste are considered. Compared to earlier models, this method produces more satisfactory products at a lower cost while producing less waste. It is found that when the precision of a process is high, it is not necessary from an economic standpoint to inspect the quality of individual components. For poor precision processes, inspecting the quality of individual components is the preferred approach from a cost/throughput standpoint.

An anomaly detection model that can detect anomalies in time series data collected from manufacturing systems is then developed. This model is based on recurrent neural networks, and it can be trained using data acquired during routine system operation. The model takes time series data as an input and reconstructs the input data. Time series data with an anomaly would causes patterns in the reconstruction errors that are inconsistent with error patterns of anomaly-free data. The performance of the proposed method is assessed using data from a diesel engine assembly process. Three common types of anomalies are detected from the time series data.

Quality control of remanufacturing is also studied. A component-oriented reassembly model is proposed. In this model, used and returned components are inspected and assigned scores according to their quality/function, and categorized in a reassembly inventory. Based on the reassembly inventory, components are paired under the control of a reassembly strategy. To evaluate the performance of different reassembly strategies under uncertain conditions, the reassembly problem is described as an agent-environment system. The platform is modeled as a Markov decision process, based on which, a reassembly-score iteration algorithm is developed to identify the optimal reassembly strategy. The effectiveness of the method is demonstrated via a case study using the reassembly process of diesel engines.

In summary, this dissertation presents a series of novel quality management models. These models are based on studying the relationship between component and product quality at multiple stages of a product life cycle. It was found that effectively managing this relationship is fundamental of improving product quality, saving cost, and reducing environmental impact. This is a novel approach to keep the competitiveness of manufacturers.

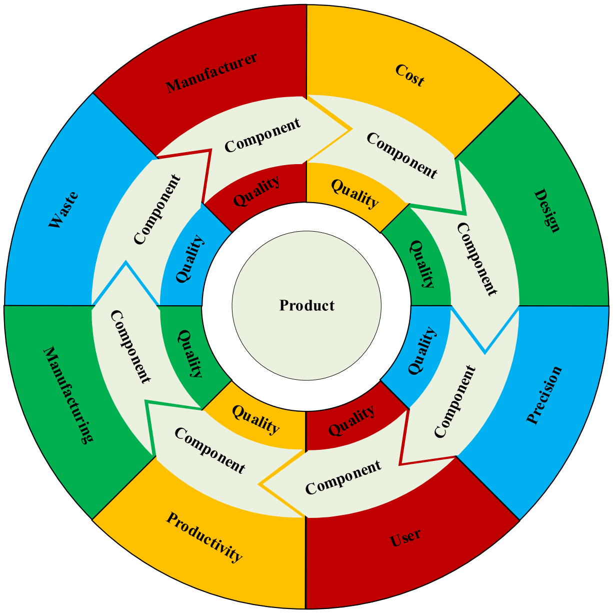


Figure 1.3. Component quality as bridges between product quality and other factors.

2. ALLOCATION OF ASSEMBLY TOLERANCES TO MINIMIZE COSTS[2]

The cost and quality of an assembly depend on the processes used to manufacture its components. The specific processes and process settings are often dictated by the tolerances on the components. One long-standing challenge is allocating the assembly tolerance to components. Many methods have been proposed, most of which endeavor to minimize cost. A tolerance allocation method that minimizes cost by jointly considering process variation and tolerance specifications is developed. This model is based on a new cost model that considers processing cost, scrap cost, and quality loss. The cost is minimized by a heuristic strategy. An overrunning clutch assembly case study is used to evaluate the method.

2.1 Introduction

Products are usually designed based on functional requirements (strength, durability, reliability, etc.). To ensure these performances, nominal values are specified. As variation in manufacturing is unavoidable, products cannot exactly meet the designed nominal values [82]. Product designers assign tolerances to the quality characteristics of the assembly. Since products are usually assembled from components, the assembly tolerances are allocated to these individual components.

Generally, the tighter the tolerance allocated to individual components, the better the function expected of the assembled product. However, components with tight tolerances must

---

[2] Reprinted (portions enhanced/adapted) from Y. Wang, L. Li, N. W. Hartman, and J. W. Sutherland, "Allocation of assembly tolerances to minimize costs," CIRP Ann., vol. 68, no. 1, pp. 13–16, 2019. https://doi.org/10.1016/j.cirp.2019.04.027. Published by Elsevier Ltd on behalf of CIRP.

typically be manufactured using machines/processes with high precision, which increases costs. Thus, there is a trade-off that must be made between quality and cost.

When a bilateral tolerance is applied to a quality characteristic, it creates lower and upper specification limits. The variation or precision of a process (characterizable by σ^2) is inversely related to the processing cost. Excessive process precision may inflate processing costs, while inadequate process precision may increase the number of products with low quality.

To address the issues noted above, a tolerance allocation method is proposed that considers both alternative processes and processing conditions. A cost model is built to address processing cost, scrap cost, and quality loss.

To minimize the cost, tolerance (more correctly referred to as variation) stack-up must be considered. As shown in Figure 2.1, manufactured quality characteristics of individual components may be viewed as stochastic variables that follow statistical distributions [82]. Component uncertainties will stack-up in the assembled products.

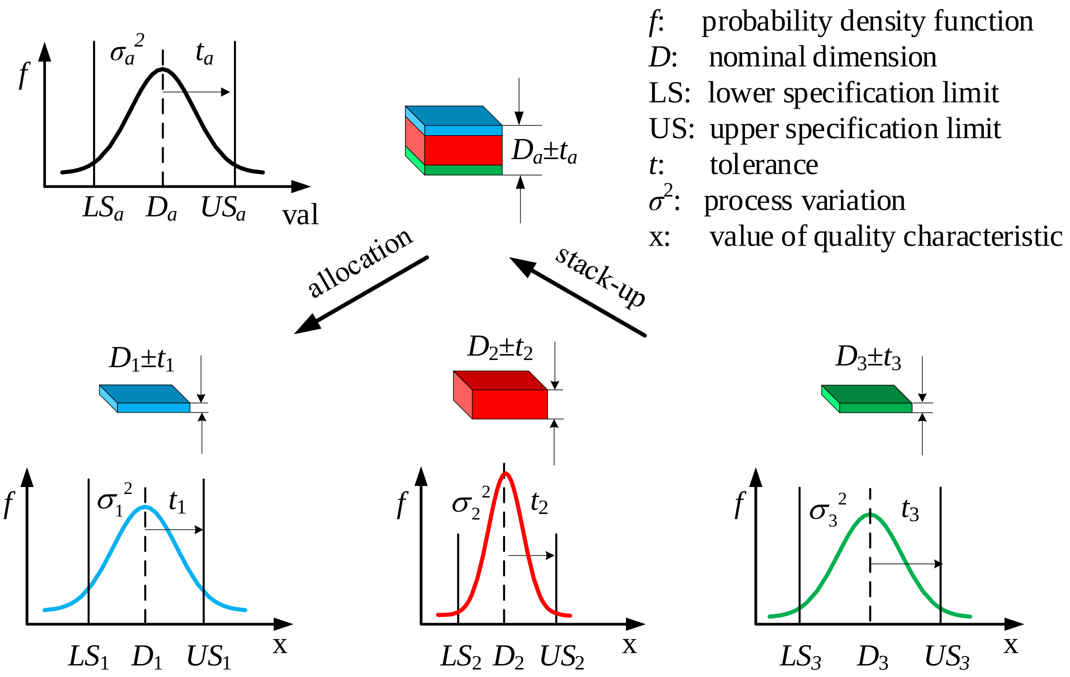


Figure 2.1. Tolerance allocation and tolerance/variation stack-up in assembly.

Tolerance allocation, when there is component variation stack-up, is often addressed using simple approaches such as worst-case model or root sum square model. Both models have limitations [19] [83]. Tolerances allocated using the worst-case model tend to be tighter than necessary, and lead to high manufacturing costs. In the root sum square model, the variance of the product's quality characteristics is assumed to be equal to the sum of the variances of the components' quality characteristics. In many cases, this assumption is not realistic. Monte Carlo simulation is well suited to predicting the distribution of products assembled from components drawn from statistical distributions [84] [85].

In this chapter, a Monte Carlo simulation is used to analyze the stack-up effect of tolerances (variations) in an assembly caused by process variations and changes in the tolerance specifications. Optimal tolerance allocation for an assembly is modeled as a discrete, nonlinear optimization problem. A heuristic strategy is proposed to solve the problem. A case study shows that this method reduces costs compared with a traditional approach based on Taylor series expansion and a loss function.

2.2 A Cost Model for Tolerance Allocation in Assembly

A cost model for tolerance allocation should include expenses from a variety of different sources. The total cost for a batch of product assemblies, C_T, can be expressed as

$$C_T = C_P + C_S + C_L, \tag{2.1}$$

where, C_P, C_S, and C_L are processing cost, scrap cost, and quality loss, respectively.

It is to be noted that a product assembled with satisfactory components may still fall outside the specifications. Components/products with quality characteristics outside the specifications are deemed unsatisfactory and will be scrapped with an associated cost. For each type of component, a constant number of satisfactory components, Q, must be delivered for assembling. The number of satisfactory products assembled is noted as M. Since some assembled products may not meet the specifications, M is equal to or less than Q.

The average cost per satisfactory product, U, can be used as the metric to evaluate the allocated tolerances [11]. U may be computed as:

$$U = \frac{C_T}{M}\,. \tag{2.2}$$

The processing cost, scrap cost, and quality loss are described in the following sub sections.

2.2.1 Processing Cost and Process-based Analysis

The processing cost of components (including unsatisfactory components that will be scrapped), C_P, can be computed by:

$$C_P = \sum_{i=1}^{m} N_i C_i\,. \tag{2.3}$$

in which, C_i is the unit processing cost of the ith component type and m is the number of different types of components assembled into one product. N_i is the number of ith components that must be manufactured to produce Q satisfactory components. N_i is given by:

$$N_i = \frac{Q}{\gamma_i}, \tag{2.4}$$

where γ_i is the probability that the ith component meets its specifications, and is given by

$$\gamma_i = \int_{LS_i}^{US_i} f_i(x)dx\,, \tag{2.5}$$

where, $f_i(x)$ is the probability density function for the ith component's quality characteristic, and LS_i and US_i are the lower and upper specification limits for the component.

Usually, product designers assign tolerances to components based on functional requirements. Process designers often divide these tolerances by a constant, k (perhaps set to 3), to obtain the standard deviation, σ. The value for σ generally dictates which processes may be used and may also indicate allowable values for processing conditions.

Many studies have modeled the processing cost as a function of tolerance, noted as $C(t)$ [4] [5]. However, it is not the tolerance, but rather, the σ that is associated with a process. With this in mind, in this section the production cost is modeled as a function of σ. This serves to establish a connection between the product design and product manufacturing. Based on this discussion, a σ-cost model for the ith component type is shown in the following equation:

$$C_i = C_i(t_i) = C_i(k\sigma_i)\,. \tag{2.6}$$

Figure 2.2 provides examples of σ-cost models. Figure 2.2 (a) shows how the cost and as-manufactured quality characteristic distribution (and σ) may change for different processing conditions. Also evident is an inverse relationship between processing cost and σ. Figure 2.2 (b) shows an example of three σ-cost curves corresponding to three processes (two processes for component A and one process for component B). Optimization should be carried out to select the types of manufacturing processes and the values of σ, so that U is minimized.

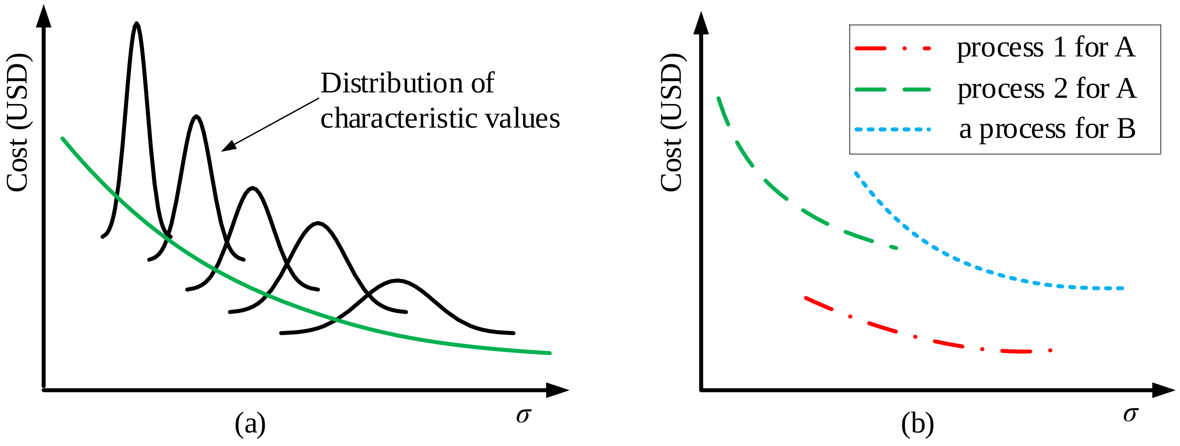


Figure 2.2. Cost and σ changes for (a) different processing conditions (b) alternative processes

2.2.2 Component Scrap Cost and Component Specification-Based Analysis

For the ith type of component, the lower and upper specification limits, LS_i and US_i, are given by (2.7) and (2.8):

$$LS_i = D_i - k\sigma_i \,, \tag{2.7}$$

$$US_i = D_i + k\sigma_i \,, \tag{2.8}$$

where D_i is the nominal quality characteristic of the component. Instead of being fixed at given values, specifications of components may be optimized (actually, the value for k is optimized), as suggested by Taguchi [86] [87]. The rationale behind undertaking this optimization is that a fixed value of k (usually 3) is not necessarily optimal. A small k will increase the scrap cost of components, while a large k will increase the possibility of assembling low quality components into products, which may increase the scrap cost of products and quality loss. A trade-off can be made by optimizing k. Since the processing cost of each component is different, the optimal value of k may also be different for each component.

It may be that different values for k can further decrease the average cost based on optimized values of σ from process-based analysis. Moreover, optimizing k can also be used independently when the relationship between σ and processing cost is not available, and σ is fixed at a given value.

An example of how component specification limits affect tolerance allocation and scrap cost is shown in Figure 2.3. Two processes with different precisions (process 2 is more precise) may be used to produce two types of components, A and B. If the tolerance on component B for process 1 is increased from t_{1B} to t'_{1B}, the probability, γ_i, that the ith component meets its specifications will increase. Thus, more B components will be considered satisfactory and be assembled. Even though the cost of scrapping unsatisfactory components might decrease, the total scrap cost might increase because more unsatisfactory products might be assembled (which are then scrapped). Note that only the component specifications are optimized, the product specifications are fixed by the design.

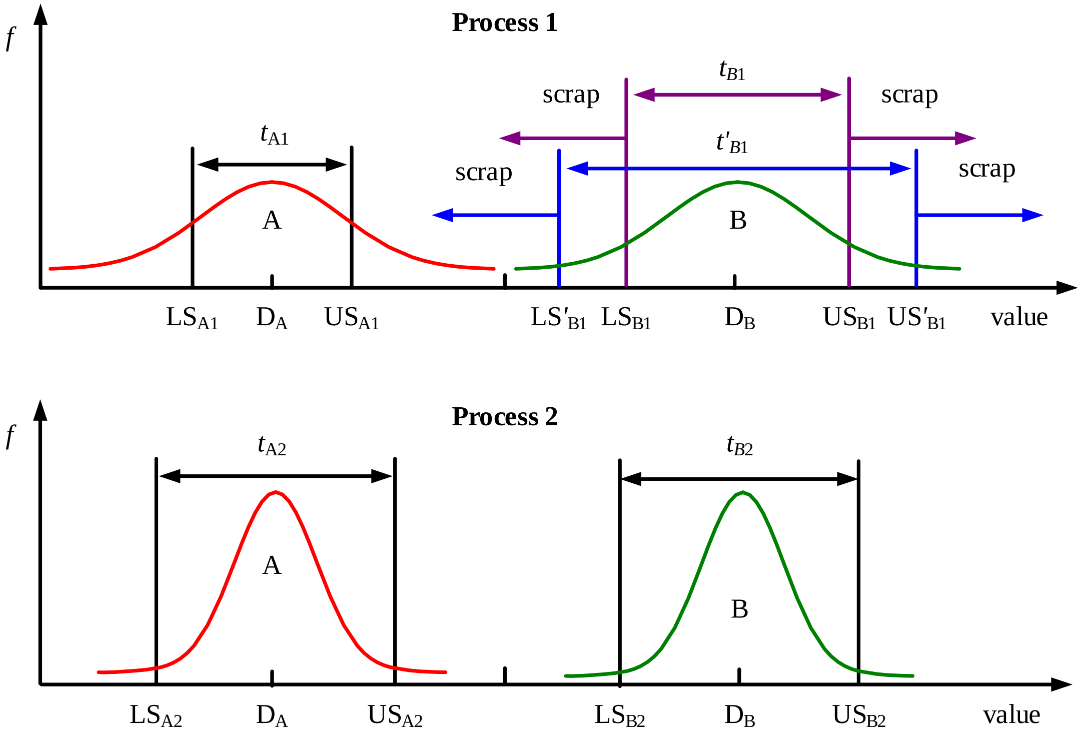


Figure 2.3. Optimize specifications

2.2.3 Quality Loss

The best product function should occur when a product quality characteristic is at the nominal value. When a quality characteristic deviates from the nominal value, product function degrades. If customers are dissatisfied with functional performance, and/or poorer function leads to product failure, loss will result (e.g., warranty costs and erosion of market position). One way to consider such broader economic costs is by introducing the loss function of a product. The loss function expresses the monetary loss due to departure of a product quality characteristic from the design nominal [86] [87]. The loss function establishes a criterion that promotes adherence to the nominal value. The most widely used loss function is the quadratic loss function (see equation (2.9)):

$$C_L = L(D_a) = \frac{A}{{t_a}^2}(D_a - D_{a0})^2 , \tag{2.9}$$

where, D_a and D_{a0} are the as-manufactured quality characteristic and nominal value for the assembled product. A is the sum of all losses associated with a product when the product quality characteristic equals the upper or lower specification limit [12], and t_a is the distance between the upper specification limit and nominal value. If a product falls within the specification limits, a societal loss is incurred; if a product falls outside the specification limits, a scrap cost is incurred. Here, the “nominal is best” form of the loss function is used; other forms may also be employed.

2.3 Tolerance Optimization Strategy and a Heuristic Algorithm

The goal of optimal tolerance allocation is achieved by minimizing the average cost per satisfactory product, U. This is a nonlinear, discrete optimization problem. It is difficult to solve the problem because of the variation/tolerance stack-up, especially for a complex assembly with alternative processes.

We carry out a Monte Carlo simulation to consider the variation stack-up. A numerical heuristic search strategy was utilized to select processes and find near optimal values of tolerances for individual components. This strategy can be used to allocate tolerances for nonlinear assembly problems when multiple processes for each component are available.

The heuristic consists of two stages: i) process-based analysis and ii) specification-based analysis. For the process-based analysis, the processes are chosen and the value of σ_i is optimized, while the constants k_i for the components are fixed at 3. For the specification-based analysis, the process types are given (values of σ_i are fixed), and the value of k_i for each component is optimized.

An overview of the algorithm to optimize σ is given below. Note that m is the number of different types of components assembled in one product, and n_i is the number of available processes that can be used to realize the ith component. Note $\sigma_{i,j}$ is the precision for the jth process of the ith component, $j \in \{1, 2, \dots n_i\}$.

1) assign process index, j, to 1 for each type of component. Set $\sigma_{i,j}$ equal to lower end of precision (σ) range as shown in Figure 2.2 (a);
2) iterate through all values of σ up to upper end of precision range. Select the σ that minimizes the value of U;
3) for ith component, consider all possible values for j,
4) perform steps 2) and 3) for each component (one at a time) until the maximum number of iterations or U converges.

The structure of the algorithm to optimize k_i parallels that of the algorithm above.

2.4 An Overrunning Clutch Assembly Case Study

To validate the proposed tolerance allocation method, an overrunning clutch assembly is analyzed. This problem is adapted from Greenwood and Chase [19] and Choi et al. [12]. A quality characteristic of the overrunning clutch performance is the contact angle, the nonlinear form of which is given by equation (2.10):

$$y = g(x_1, x_2, x_3, x_4) = \cos^{-1}\left(\frac{x_1 + (x_2 + x_3)/2}{x_4 - (x_2 + x_3)/2}\right), \tag{2.10}$$

where, y is the contact angle, x_1 is the diameter of the hub, x_2 and x_3 are the diameters of the rollers, and x_4 is the inside diameter of the cage. The nominal dimensions of x_1, x_2, x_3, and x_4 are 55.29 mm, 22.86 mm, 22.86 mm, and 101.69 mm, respectively, as shown in Figure 2.4. For proper function, the contact angle, y, should be constrained between 0.122±0.035 rad (7.0±2.0 deg).

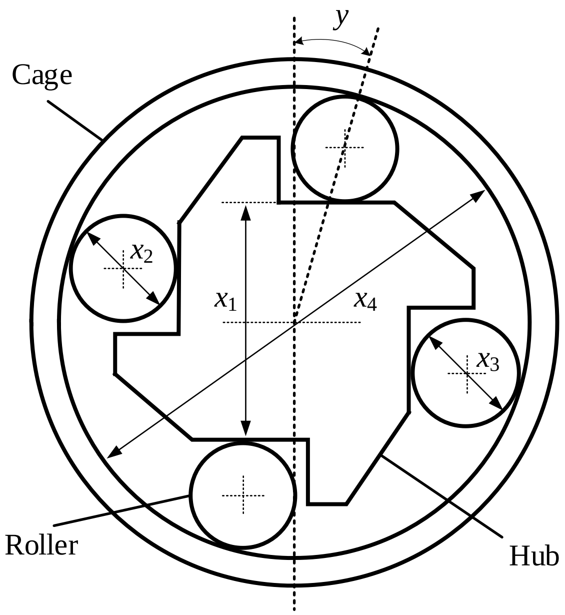


Figure 2.4. Overrunning clutch assembly [12]

In the case study of Choi et al. [12], a reciprocal tolerance-cost function was used. Employing our notation, this becomes:

$$C_i(k_i\sigma_i) = a_i + \frac{b_i}{k_i\sigma_i}, \tag{2.11}$$

where a_i is the fixed cost related to tooling, setup, etc., and $b_i/k_i\sigma_i$ is the variable cost of processing a component with the given tolerance.

Each component has alternative processes, and each process has a tolerance range. For comparison purposes, the constants (a_i and b_i) for the alternative processes, the tolerance ranges, and the values for A in the product quality loss function are those used by Choi et al. [12].

The sample size for the Monte Carlo simulation was 10,000. The scrap cost of each unsatisfactory component was assumed to be 10% of a_i. The scrap cost of each unsatisfactory product was \$1.5, which was about 8 percent of the cost of producing a new product.

2.4.1 Process-based Analysis

In this section, the process for each component was chosen and the value of σ was optimized while the tolerance constant k was fixed at 3. The value ranges for σ were calculated by dividing each tolerance range given in Choi et al. [12] by 3.

The results using the methods of Choi et al. [12] and the methods proposed in Chapter 2.3 are shown in **Error! Reference source not found.**. The two methods selected the same process t ypes. When A equals 0 or 20, the proposed method allocated a larger σ (looser tolerance) to each component. Even though fewer satisfactory products were assembled, a lower average cost per satisfactory product, U, was achieved. When the loss constant A equals 100, the proposed method allocated a smaller σ (tighter tolerance) to each component, and assembled more satisfactory products with a smaller value of U. It seems that the proposed method reduced average cost by avoiding excessive precision (σ) and by more effectively allocating tolerances among individual components.

2.4.2 Specification-Based Analysis

In this section, the values of k for each type of component are optimized while the process type and σ for each component remain fixed at the values obtained in Chapter 2.4.1. In the iteration process, the upper bound for each k is 6. Results using the heuristic algorithm proposed in Chapter 2.3 are shown in **Error! Reference source not found.**.

From the table it can be concluded that adjusting the value of k_i can further reduce the value of U and assemble more satisfactory products. The specifications for components with lower processing cost (inexpensive components) were tightened, while the specifications for components with higher processing cost (expensive components) were loosened. This reduces the loss caused by assembling expensive components into products that will be scrapped because of large variance stack-up, which is mainly accumulated from the variance of inexpensive components.

Table 2.1. Results of process-based analysis

	Choi et al. [12]									This study								
	A=0			A=20			A=100			A=0			A=20			A=100		
	Proc. index	σ	*N-Q*	Proc. index	σ	*N-Q*	Proc. index	σ	*N-Q*	Proc. index	σ	*N-Q*	Proc. index	σ	*N-Q*	Proc. index	σ	*N-Q*
x_1	3	0.059935	24	3	0.057957	20	2	0.045559	29	3	0.083033	19	3	0.083033	31	2	0.049700	22
x_2	2	0.055119	30	2	0.055415	31	2	0.049850	24	2	0.096400	21	2	0.078100	30	2	0.049900	18
x_3	1	0.040044	20	1	0.042796	31	1	0.038527	28	1	0.063967	24	1	0.056167	33	1	0.043867	29
x_4	3	0.066860	32	3	0.066667	36	3	0.066667	35	3	0.128533	23	3	0.089533	29	3	0.067333	27
C_L	0			32352			146580			0			46862			143730		
M	9950			9954			9978			9259			9724			9980		
U	24.68			27.90			40.83			20.26			25.84			39.85		

Table 2.2. Results of specification-based analysis

	A=0			A=20			A=100		
	k	C_i	*N-Q*	*k*	C_i	*N-Q*	*k*	C_i	*N-Q*
x_1	5.48	6.51	0	2.45	6.51	241	5.06	8.35	0
x_2	3.16	5.25	22	2.31	5.77	0	5.40	7.34	2
x_3	3.25	4.06	11	2.49	4.28	0	2.16	4.78	398
x_4	1.78	2.78	1106	1.79	3.78	832	1.59	4.86	1774
C_L	0			42190			129988		
M	9450			9841			9996		
U	20.02			25.58			39.09		

2.4.3 Summary and Conclusions

This study has proposed a method for tolerance allocation that minimizes costs. The cost model considered processing cost, scrap cost, and quality loss. By optimizing the precision of manufacturing processes and width of tolerance specifications, a trade-off was made among costs of production, scrap, and quality loss. The tolerance allocation problem was modelled as a nonlinear, discrete optimization problem. A Monte Carlo simulation was used to analyze the variation/tolerance stack up. A heuristic search strategy was proposed to find solutions. In a case study, a traditional method was compared to the proposed model. The results showed that the proposed model decreases the average cost by avoiding unnecessary process precision, more effectively allocating tolerances among individual components, and optimizing tolerance specifications.

The proposed method approaches the tolerance allocation problem by optimizing both the precision of manufacturing processes (σ) and the specification limits (k) of components. This establishes a connection between product design and product manufacturing. If specification limits of expensive components are loosened, and specification limits of inexpensive components are tightened, a lower average cost per satisfactory product can be achieved. As is evident from the case study, the proposed approach provides a superior solution to the tolerance allocation problem.

3. TOLERANCE ALLOCATION: BALANCING QUALITY, COST, AND WASTE THROUGH PRODUCTION RATE OPTIMIZATION[3]

Dimensional tolerance allocation is a very important and difficult task that traditionally seeks to balance cost/productivity and quality. Common tolerance allocation models have two shortcomings: i) they are overly reliant on models focused on minimizing cost and tend to ignore waste, and ii) they fail to connect to the root cause of many quality issues: process variation. This chapter proposes a tolerance allocation model that addresses these shortcomings. The proposed model considers both product design (tolerance selection) and operation planning (or production rate selection). Relations among production rate, production cost, processing precision, and waste are considered. A gradient-based optimization method is proposed to minimize the cost and waste. A clutch assembly case study is analyzed to evaluate the method. Monte Carlo simulations are employed to validate the accuracy of the proposed cost model. The proposed method is compared to the method proposed in Chapter 2. The proposed method produced more satisfactory products at a lower cost while producing less waste. For the case study, it is found that when the precision of a process is high, it is not necessary from an economic standpoint to inspect the quality of individual components. For poor precision processes, inspecting the quality of individual components is the preferred approach from a cost/throughput standpoint.

---

[3] Reprinted (portions enhanced/adapted) from Y. Wang, A. Huang, C. A. Quigley, L. Li, and J. W. Sutherland, "Tolerance allocation: Balancing quality, cost, and waste through production rate optimization," J. Clean. Prod., vol. 285, p. 124837, 2021. https://doi.org/10.1016/j.jclepro.2020.124837. Published by Elsevier Ltd

3.1 Introduction

Table 3.1. Nomenclature.

β	Product pass rate	C_S (\$)	Total scrap/recycle cost
γ	Component pass rate	C_T (\$)	Total production cost
δ	Design function sensitivity	E (mm)	Process related constant
μ	Process mean	F (mm·min^2)	Process related constant
σ	Process standard deviation	L	Number of scrapped components
k	Tolerance spread	LS	Lower specification limit
r	Production rate	M	Number of satisfactory products
t	Tolerance	N	Number of components processed
x	Characteristic value of a component	Q	Number of components assembled
y	Characteristic value of a product	R_N	Remainder of Maclaurin series
A (\$)	Fixed cost (set-up cost)	S_C (\$)	Scrap/recycle cost of a component
B (\$/min)	Cost coefficient	S_N	Maclaurin series
B_E (\$/min)	Electricity cost	S_P (\$)	Scrap/recycle cost of a product
B_L (\$/min)	Labor cost	U (\$)	Average unit cost of a product
B_M (\$/min)	Machine tool cost	US	Upper specification limit
C_B (\$)	Processing cost	W	Number of unsatisfactory products

The ability to produce high quality products with low cost and high production rate is critical for manufacturers. In addition, waste from manufacturing has become a severe environmental burden, thus, reducing the production of waste streams and efficient use of material resources in manufacturing is ever more important from an environmental sustainability perspective [88]. Usually, high quality (precision), low cost, low waste, and high production rate are conflicting objectives, because excessive precision leads to excessive cost and processing time [27] [89]. For example, to achieve high precision, a larger investment must be made in purchasing and maintaining highly precise machine tools and maintaining them more carefully at an associated higher cost (e.g., changing the tooling more frequently) [90]. In addition, more precise machine tools are generally more complex and precise metrology equipment, which is more expensive. On the other hand, when inexpensive processes with low precision are applied, component-to-component variation will be large, and the resulting quality of assembled products may not meet the expectation of customers [91]. Also, when variation is large, components/products outside the specifications are likely rejected, and, if they cannot be reworked or recycled, they are scrapped (i.e., enter the waste stream), which is also wasteful in terms of energy consumption and resource depletion.

Owing to the ever increasing requirements of high-quality, low-cost products, and the awareness of sustainability, more and more efforts have been carried out to study the relations among product quality, cost, and waste reduction [81], [92]–[96]. This chapter extends this effort by providing a tolerance allocation model that balances quality, time, cost, and waste through the optimization of production rate.

Tolerances are assigned for critical product/component characteristics during the engineering design process. Tolerance is the amount by which a characteristic value is allowed to deviate from the nominal value; it acknowledges that a manufacturing process cannot exactly realize a nominal value [97]. Most often, tolerances are selected based on product/component function considerations, as well as quality and cost. The tolerance on a component is used by manufacturing planners to select appropriate processes and their sequences (process planning) and the settings and tooling for each process (operation planning) [98] [99].

The tolerancing problem in engineering design has been widely studied by transforming it into a constrained optimization problem. The most common way of formulating such a problem is to establish allowable tolerances on a product based on functional considerations, e.g., the clearance between a shaft and hole must not be too small (this may inhibit assembly) and it must not be too large (this may not provide sufficient sealing). While the tolerances on a product are likely based on functional considerations and cost, since products are composed of components a designer must address the issue of how to allocate product tolerances to the components.

Traditional tolerance allocation methods allocate tolerances using a product/component design-oriented approach, in which almost all the focus is placed upon the tolerances of the product/components. Product engineering uses “tolerance” to communicate what is acceptable in terms of function. Manufacturing decision-makers must translate “tolerance” into their language because they think in terms of process variation or precision. The authors believe that rather than a single-minded focus on tolerance, that it is better to think in terms of process variation, since the same process settings that influence variation also determine cycle time, production cost, and amount of waste.

Few studies on tolerancing that consider the linkage between design and production have been carried out. For the few studies that have considered this linkage, attention was devoted to relating process variables and process variation for a specific situation – the connection among process variables, process variation, and tolerance was not considered in a broader sense. It is a relatively simple matter to describe how different conditions/variable settings affect the production rate for an operation, and production rate is strongly linked to the variation/precision and cost of an operation [100]. A planner has the latitude to explore different production rates (e.g., by adjusting process types and process variables), so long as the required tolerance is met [100]. For a given tolerance and associated process, if the maximum allowable production rate does not produce high enough throughput, or has too high a cost, then the tolerance will need to be loosened. This begins to illustrate the types of trade-offs that need to be made among quality, cost, and production rate [101]–[104].

To fill the research gap summarized above, a new tolerance allocation method is proposed. This method avoids the three problems of traditional tolerance allocation methods. One, not only cost, but also waste is minimized. This method avoids both unnecessary excessive precision and a large level of product/component scrap caused by quality or assembly issues. Two, instead of considering how to allocate a product tolerance by minimizing the sum of the costs associated with individual operations by changing process parameters, this chapter optimizes the production rate for operations to achieve the product tolerance. Optimizing production rate makes the model easy to be generalized to different operations. Three, the linkage between design and production are considered. This is achieved by using statistical theory to characterize the relations among production rate, process precision, and process/component variation stack-up. A gradient-based optimization method is proposed to find optimal production rates that satisfies the tolerance requirement on a product and minimizes cost and waste. An overview of the research is shown in Figure 3.1.

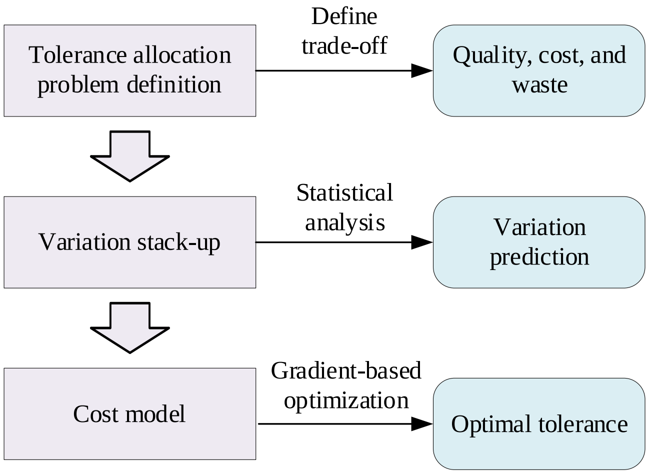


Figure 3.1. Overview of this research.

3.2 Problem Description and a Cost Model

This chapter considers a tolerance allocation problem, i.e., how to best allocate the tolerance on a product assembly to the individual components. The tolerance of a component can be defined as the distance between the nominal value and the upper/lower specification limit. For a symmetric bilateral case, where the upper and lower specification limits have the same distance from the nominal value, the tolerance, t, is given by equation (3.1):

$$t = x_0 - LS = US - x_0 = k\sigma\,, \tag{3.1}$$

where, LS is the lower specification limit, US is the upper specification limit, x_0 is the nominal value, σ is the process standard deviation, and $k\sigma$ is the size of the tolerance (k will be referred to as the *tolerance spread*).

The tolerance allocation challenge can be illustrated by a shaft-hole assembly example, as shown in Figure 3.2. It is assumed that only products/components with characteristic values between the lower/upper specifications, noted as satisfactory products, can be sold. Unsatisfactory products/components lying outside the specifications will be scrapped and managed as waste. The ratio between the number of satisfactory products and the total products assembled is the product pass rate, β. The shaft-hole clearance, i.e., characteristic value of the assembly, y, is the difference between the diameter of the shaft, x_s, and the hole, x_h. The diameters of the shaft and hole are assumed to be normally distributed and are centered at the nominal values, x_{s0} and x_{h0} (the nominal value for the shaft/hole clearance is $y_0=x_{h0}-x_{s0}$). The tolerance for the clearance is t_y, which should be allocated to the tolerances on the diameters of the shaft, t_s, and hole, t_h. Based on the allocated tolerances, manufacturing planners select appropriate processes settings and tooling to produce the shaft and hole. It is assumed that random assembly is employed, i.e., a shaft and a hole are each randomly selected and assembled. The shaft and hole each have a probability distribution associated with their size, and based on random assembly, the clearance also has a distribution (hole and shaft stack-up). In allocating the clearance tolerance to the tolerances on the shaft and hole diameters, statistical methods are needed to predict how the variations in the shaft and hole creation processes influence the clearance variation.

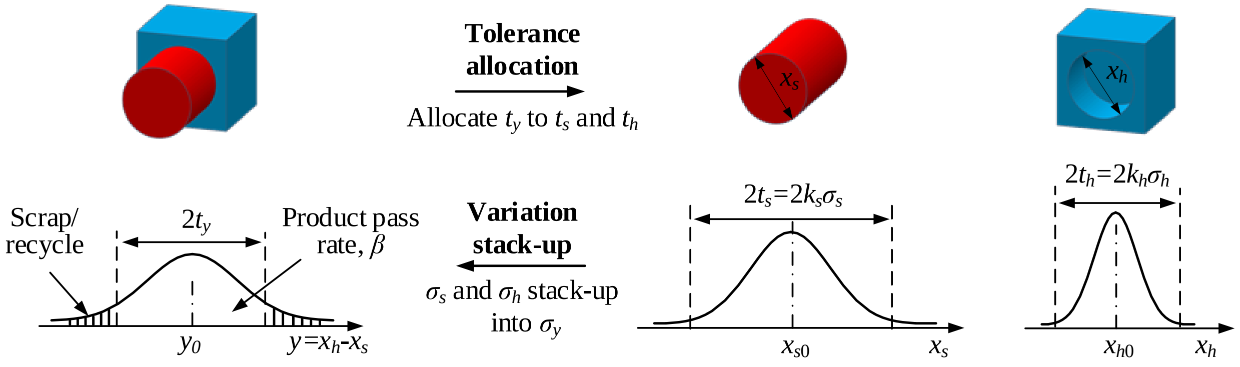
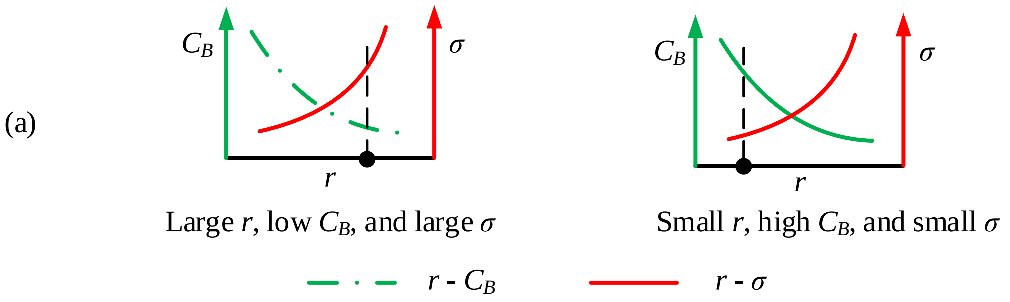


Figure 3.2. A shaft and hole tolerance allocation example.

One strategy that might be considered by a manufacturer would be to use low precision processes to fabricate the components, which would keep manufacturing costs low. Then, a tight tolerance could be applied to the components to filter out poor quality components (this would lead to higher scrap costs/waste). This strategy would probably lead to a relatively high proportion of assembled products that satisfy the product specifications (less scrap products waste). Alternatively, process type and process parameter settings could be used to achieve precise processes, but this would come with a higher cost. However, this would likely lead to less scrap components/waste. In general, tolerances on the components should be set with an overall consideration of precision, cost, and waste.

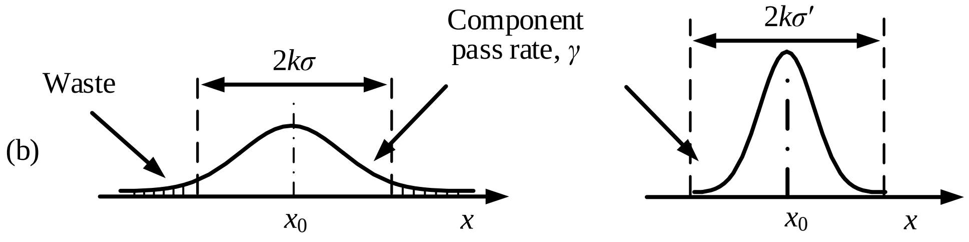
The production rate for a process depends on the type of process and the condition/variable settings for the process. The production rate in turn impacts the precision and cost of an operation [100]. Many studies have been carried out to study the relation between production rate, product quality, and processing cost [102] [103]. General relations between production rate (r), processing cost (C_B), and process standard deviation (σ) are shown in Figure 3.3 (a) and (b). It may be noted that as the process standard deviation, σ (or variation, σ^2) increases, the process precision erodes. Generally, the higher the production rate (usually achieved by increasing a process variable such as feed rate, step over, and cutting depth), the lower the processing cost. This is the case, since for high production rates more components are produced per unit time, and thus fixed costs are allocated across more components. However, many studies have shown that when a high production rate is applied, component-to-component variation is larger, i.e., quality and precision are decreased [105] [106]. For example, consider a milling process; when a large feed rate is used, the quality of the machined part (evaluated by a criterion such as surface roughness or form error) will decrease [107]. This work will propose a tolerance allocation method that optimizes the production rate, r, so as to balance quality, cost, and waste.

Many in industry use inspection-oriented approaches to identify and remove out of specification component to ensure quality. The tolerance and nominal value together serve to define the specification limits. During inspection, components within the specification limits are deemed satisfactory, and can be assembled. Otherwise, the components are rejected and scrapped (where they enter the waste stream or are recycled). Both the tolerance and process precision affect the component pass rate, γ, which is the ratio between the number of satisfactory components and the total number of manufactured components, as shown in Figure 3.3 (b) and (c). The precision of the process depends on the production rate, with higher rates generally leading to reduced precision. The often utilized $\pm 3\sigma$ tolerance band may also be modified to adjust the pass rate; herein values other than “3” for k (tolerance spread) are considered. k may be adjusted to avoid unnecessary scrap or to avoid passing too many low-quality components, as shown in Figure 3.3 (c).



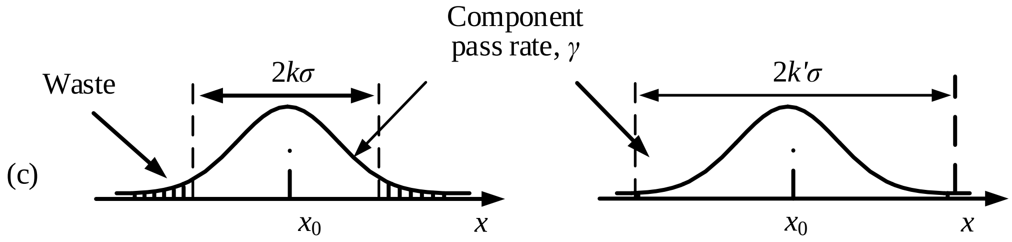
Large r, low C_B, and large σ Small r, high C_B, and small σ

Relations among r, C_B, and σ



Large σ (large r), low C_B Small σ (small r), high C_B

Adjust σ by changing r



Small k, low pass rate γ Large k, large pass rate γ

Adjust k

Figure 3.3. The relation among production rate (r), processing cost (C_B), precision (σ), and tolerances spread (k)

To find a trade-off among quality, cost, and waste, an optimal tolerance allocation method has to consider three factors: (1) the impact of production rate on the cost, variance of components, and waste caused by unsatisfactory components/products, (2) the stack-up of component variations for an assembled product, and (3) different quality management strategies, and their impact on cost and the quality of products/components. The average unit cost of a satisfactory product assembled, U, will be used to evaluate the economic and environmental performance (since the cost of waste is considered) of the production system. This work builds a cost model to calculate U, with these three factors considered.

To understand the cost model, it is necessary to define the relations between the number of products and components, under different quality inspection strategies. Consider an assembly process, in which m types of components are assembled into a product. For each component type, Q components are provided for assembly. All the products are assembled and are then inspected. The number of unsatisfactory products is W, and the number of satisfactory products is M. The summation of W and M equals to Q. For random assembly, a product that is assembled from satisfactory components may still have a characteristic value that falls outside its tolerance.

Three common component inspection strategies are considered: i) no inspection, ii) 100% inspection, and iii) acceptance sampling. If no inspection is carried out, all components are assembled. If 100% inspection is carried out, every component is inspected, and only the satisfactory components will be assembled, and the unsatisfactory components will be scrapped/recycled [108]. For acceptance sampling, inspection is carried out on a small subset of the components. If the qualities of the sampled components are acceptable, then all the components will be considered acceptable, and will be assembled. But if the subset of components is deemed to have poor quality, then 100% inspection is performed on all the components.

For both the case of no inspection and the case of acceptance sampling (when the sample of components pass inspection), the vast majority of components are not inspected. These two cases will serve as scenario one: “no component inspection.” For the case of 100% inspection and the case of acceptance sampling (when the sample of components fails inspection), every component is inspected. These two cases will serve as scenario two: “100% inspection.”

The assembly processes for both scenarios are shown in Figure 3.4, which is illustrated by a product assembled from two types of components. In scenario one, there will be no waste associated with scrapping/recycling components. For each component type, the total number of components that are processed is equal to Q. In scenario two, every component is inspected. Only satisfactory components will be assembled, while unsatisfactory components are scrapped. For the ith component type, N_i components are manufactured, with L_i not meeting the specifications and Q meeting the specifications. For both scenarios, the quality of every product is inspected. The numbers of satisfactory product and unsatisfactory products are M and N.

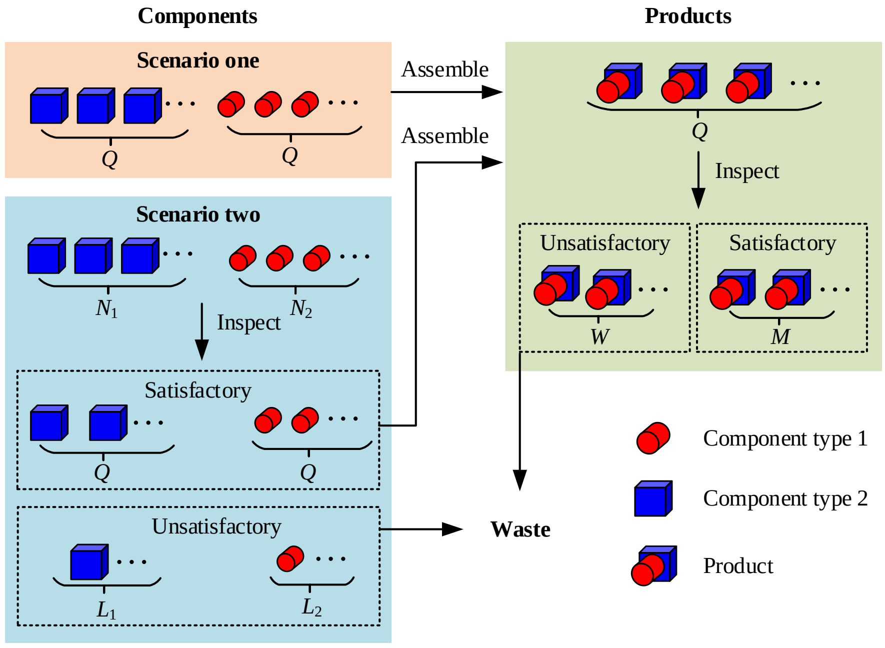


Figure 3.4. An assembly problem.

The average unit cost, U, of a satisfactory product, can be calculated as:

$$U = \frac{C_T}{M}, \tag{3.2}$$

where, M is the number of satisfactory products assembled. C_T is the total cost, which includes the costs incurred in manufacturing and assembling all the components and managing the scrap. C_T is defined as:

$$C_T = C_B + C_S, \tag{3.3}$$

where, C_B is the total processing cost of all the components, and C_S is the scrap/recycle cost of all the unsatisfactory products and unsatisfactory components.

The total processing cost, C_B, is the summation of processing cost of all m types of components assembled into the product, and is given by the following equation:

$$C_B = \sum_{i=1}^{m} C_{Bi} N_i, \tag{3.4}$$

where, i is the component type index, N_i is the number of components of type i that are manufactured, and C_{Bi} is the processing cost per unit of component i.

The total scrap cost, C_S, is

$$C_S = WS_p + \sum_{i=1}^{m} L_i S_{Ci}, \tag{3.5}$$

where, S_P is the cost to scrap an unsatisfactory product, W is the number of unsatisfactory products, S_{Ci} is the cost to scrap the ith component, and L_i is the number of unsatisfactory components of type i.

3.3 Methodology

In this section, a tolerance allocation method that minimizes the average unit cost is proposed. The influence of production rate on process precision and cost is considered. This method avoids both unnecessarily high and unacceptable low process precision. Two assumptions are made for the tolerance allocation problem. First, the process is under statistical control (i.e., the process mean and variation are assumed stable), and the characteristic value of a component can be modeled as a random variable that follows a normal distribution, with the mean, μ, being equal to the design-specified nominal value, x_0 [87] [109]. Second, there are no constraints on the production time, i.e., for a specified production rate there is sufficient time to produce the required number of components [13].

The component processing cost, C_B, can be modeled as a function of production rate, r_i:

$$C_B = h(r_i)\,. \tag{3.6}$$

Similarly, the value of σ for a component can be modeled as a function of r:

$$\sigma = g(r_i)\,. \tag{3.7}$$

Equations (3.6) and (3.7) provide general functional forms for the cost and precision of a process. Some studies have provided general forms of the equations [103]. In Section 3.4.2, expressions are presented for these two equations.

Figure 3.5 summarizes how the production rate, r, and the tolerance spread, k, affect the average unit cost, U, through a diagram. The relations among these variables are given in Equations (3.2)-(3.7). For scenario one, no inspection is carried out for the components. Thus, tolerances are not needed for the components and the tolerance spread need not be defined; rather, the production rate (and thus precision, σ) for the component processes is used to control the precision. For scenario two, both production rate and tolerance spread, k, are controlled variables.

As shown in Figure 3.5, for scenario one, the processing cost, C_{Bi}, and the variance, σ_{xi}^2, of individual components is affected by the production rate for that process, r_i. The variations of individual components σ_{xi}^2 will stack-up into the variations of a product, σ_y^2, which affects the product pass rate β (when the products are inspected relative to the specifications). Both the scrap cost of products, S_P and the number of satisfactory products, M, are determined by β. For scenario two, the pass rate, γ, for a given component is determined by σ_{xi}^2 and k; the product pass rate, β, determined by σ_y^2; and the product tolerance, t_y (a fixed value by design). The tolerance spread, k, affects both σ_{xi}^2 and σ_y^2. The pass rates for the components and the product determine the total scrap cost associated with the components and product.

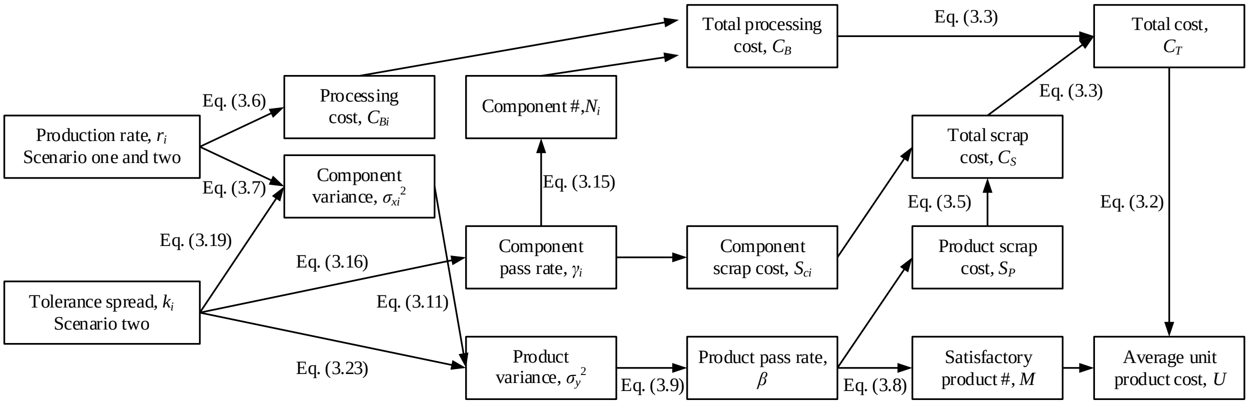

Total processing cost, C_B
Eq. (3.3)
Total cost, C_T
Eq. (3.6)
Processing cost, C_{Bi}
Component #, N_i
Eq. (3.3)
Production rate, r_i Scenario one and two
Total scrap cost, C_S
Eq. (3.7)
Component variance, σ_{xi}^2
Eq. (3.15)
Eq. (3.5)
Eq. (3.2)
Eq. (3.19)
Component pass rate, γ_i
Component scrap cost, S_{ci}
Product scrap cost, S_P
Eq. (3.16)
Tolerance spread, k_i Scenario two
Eq. (3.11)
Eq. (3.23)
Product variance, σ_y^2
Eq. (3.9)
Product pass rate, β
Eq. (3.8)
Satisfactory product #, M
Average unit product cost, U

Figure 3.5. Logic flow of the cost model including key model parameters

3.3.1 Statistical Analysis of Variation Stack-up

For a product assembled from m types of components, its characteristic value, y, is determined by the characteristic values of components assembled, noted as x_1, x_2, …, and x_m, through the design function $y=f(x_1, x_2, \ldots, x_m)$. Usually, different types of components are processed independently, so variables x_1, x_2, …, and x_m can be modeled as independent variables.

Herein, it is assumed that x_i is normally distributed. Let us also assume that y is a normally distributed random variable (because random assembly is deployed, also, this assumption is verified in the case study).

The number of satisfactory products, M, can be estimated by:

$$M = Q\beta(r_1, r_2, ..., r_m)\,, \tag{3.8}$$

where, $\beta(r_1, r_2, \ldots r_m)$ is the pass rate of products, which is a function of the production rate of each process, r_i. Since the product distribution is unbiased (i.e., the product distribution is centered at the nominal value, y_0) and a bilateral tolerance is used, the function $\beta(r_1, r_2, \ldots r_m)$ can be evaluated with Equation (3.9).

$$\beta(r_1, r_2, ..., r_m) = \int_{y_0 - t_y}^{y_0 + t_y} \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y-y_0)^2}{2\sigma^2{}_y}}\, dy = \frac{2}{\sqrt{\pi}} \int_0^{\frac{t_y}{\sqrt{2}\sigma_y}} e^{-s^2}\, ds = \operatorname{erf}\left(\frac{t_y}{\sqrt{2}\sigma_y}\right), \tag{3.9}$$

where, erf(x) is the Gauss error function, and t_y is the tolerance of the product.

The number of unsatisfactory products, W, can be computed given the pass rate of the product, as shown below:

$$W = Q[1 - \beta(r_1, r_2, ..., r_m)]\,, \tag{3.10}$$

Two options are available to predict how the variations in individual components stack-up in the product: variation simulation by Monte Carol simulation and statistical theory-based approach. Monte Carlo simulation predicts the distribution associated with an assembly as parts are randomly drawn from distributions associated with each individual component and virtually assembled. Monte Carlo simulation may lead to excessive computation time. The statistical theory-based approach relates the component variances σ_{xi}^2 to the product variance σ_y^2 using a first order approximation of the design function. This is shown in Equation (3.11) [110]:

$$\sigma_y{}^2 \approx \sum_{i=1}^{m} \delta_i{}^2 \sigma_{xi}{}^2 , \tag{3.11}$$

where δ_i is the partial derivative of y with respect to x_i (the design function sensitivity with respect to x_i):

$$\delta_i = \frac{\partial y}{\partial x_i}\Big|_{\mu_1,\mu_2,\ldots,\mu_m} . \tag{3.12}$$

In Equation (3.12) it is assumed that the design function $f(x_1, x_2, \ldots x_m)$ is differentiable. If $f(x_1, x_2, \ldots x_m)$ is not differentiable, the δ_i values may be estimated using a numerical approximation.

3.3.1.1 Scenario One: No Inspection of Components

In this subsection, scenario one is considered: none of the components are inspected (or, for the case of acceptance sampling, very few). The number of components produced, N_i, is equal to Q. The number of scrap components of type i, L_i, is equal to 0. Thus, the scrap cost of unsatisfactory components is 0, and the total scrap cost can be simplified to:

$$C_S = WS_p . \tag{3.13}$$

By placing Equations (3.4), (3.9), and (3.13) into Equation (3.2),the following expression is obtained for the average unit cost, U:

$$U(r_1, r_2, ..., r_m) = \frac{\left(\sum_{i=1}^{m} C_{Bi}(r_i)Q\right) + WS_p}{Q\beta(r_1, r_2, ..., r_m)} = \frac{\sum_{i=1}^{m} C_{Bi}(r_i) + \left(1 - \text{erf}\left(\frac{t_y}{\sqrt{2}\sigma_y}\right)\right)S_p}{\text{erf}\left(\frac{t_y}{\sqrt{2}\sigma_y}\right)}. \tag{3.14}$$

The production rate, r_i, impacts the average unit cost by affecting the processing cost and the component variation stack-up (evaluated by σ_y).

3.3.1.2 Scenario Two: 100% Inspection of Components

In this subsection, scenario two is considered: every component is inspected and judged as satisfactory/not satisfactory. The number of components of the *i*th type that must be processed to produce *Q* satisfactory components is N_i, and can be calculated using:

$$N_i = \frac{Q}{\gamma_i}, \tag{3.15}$$

where γ_i is the pass rate for component type *i*, which is affected by the tolerance spread, k_i, of the component. Similar to computing the pass rate of a product, β, the pass rate of a component, γ_i, can be evaluated using a normal distribution. Since the process is unbiased (mean of process, μ_i, is equal to the nominal value x_{i0}) and a symmetrical bilateral tolerance is used, the component pass rate can be evaluated with Equation (3.16):

$$\gamma_i = \int_{x_{i0}-k_i\sigma_{xi}}^{x_{i0}+k_i\sigma_{xi}} \frac{1}{\sqrt{2\pi\sigma_i}} e^{\frac{-(x-x_{i0})^2}{2\sigma^2{}_{xi}}} dx = \frac{2}{\sqrt{\pi}} \int_0^{\frac{k_i}{\sqrt{2}}} e^{-s^2} ds = \text{erf}\left(\frac{k_i}{\sqrt{2}}\right), \tag{3.16}$$

where, $k_i\sigma_i$ is the component specification (tolerance).

The number of components scrapped, L_i, is

$$L_i = N_i - Q = Q\left(\frac{1}{\gamma_i} - 1\right). \tag{3.17}$$

By placing Equations (3.15)-(3.17) into Equation (3.2), the average unit cost, U, can be represented as a function of r_i and k_i:

$$\begin{aligned} U(r_1,\ldots,r_m,k_1,\ldots,k_m) &= \frac{\sum_{i=1}^{m}\left(\frac{C_{Bi}(r_i)}{\gamma_i} + S_{C_i}\frac{1-\gamma_i}{\gamma_i}\right) + WS_p}{\beta(r_1,r_2,\ldots,r_m)} \\ &= \frac{\sum_{i=1}^{m}\left(\frac{C_{Bi}(r_i)}{\operatorname{erf}\left(\frac{k_i}{\sqrt{2}}\right)} + \frac{S_{Ci}}{\operatorname{erf}\left(\frac{k_i}{\sqrt{2}}\right)} - S_{C_i}\right) + \left(1 - \operatorname{erf}\left(\frac{t_y}{\sqrt{2}\sigma_y}\right)\right)S_p}{\operatorname{erf}\left(\frac{t_y}{\sqrt{2}\sigma_y}\right)}. \end{aligned} \tag{3.18}$$

Unsatisfactory components will be removed through inspection, and the resulting distribution of "passed" components will follow a truncated normal distribution. The standard deviation of the truncated normal distribution, σ_{xi}', is smaller than the standard deviation of the distribution before truncation, σ_{xi}. The value of σ_{xi}' can be computed from σ_{xi} using the following expression [87]:

$$\sigma'_{xi} = h(k_i)\sigma_{xi}, \tag{3.19}$$

where, $h(k_i)$ is a function of k_i, and is given by:

$$h(k_i) = \sqrt{1 - 2k_i T(k_i)\varphi(-k_i)}, \tag{3.20}$$

and $T(k_i)$ and $\varphi(k_i)$ are functions of k_i, and can be expressed as:

$$T(k_i) = \frac{1}{\operatorname{erf}\left(\frac{k_i}{\sqrt{2}}\right)}, \tag{3.21}$$

$$\varphi(k_i) = \frac{1}{\sqrt{2\pi}} e^{\frac{-k_i^2}{2}}. \tag{3.22}$$

The function $h(k_i)$ is shown in Figure 3.6. When k_i (tolerance spread) increases, meaning that a larger tolerance is assigned and fewer components are scrapped, then σ_{xi}' also increases and approaches σ_{xi}. When k_i decreases, a tighter tolerance is applied and more components are scrapped, and thus the value of σ_{xi}' decreases.

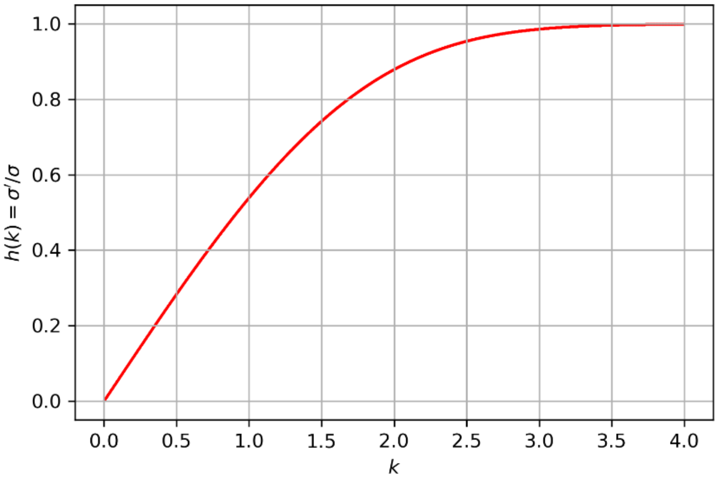


Figure 3.6. Behavior of $h(k)$, ratio of standard deviation of truncated normal distribution to standard deviation of starting distribution.

Figure 3.5 shows how the values of k_i for the individual components affects their respective standard deviations. The collective effect of these k_i values on the standard deviation of the assembled products is given by the following function:

$$\sigma_y \approx \sqrt{\sum_{i=1}^{m}\left(\delta_i h(k_i)\sigma_{xi}\right)^2}\ . \tag{3.23}$$

Furthermore, let us assume that the shape of the distribution of the characteristic value of the product, y, is not too dramatically affected by truncations in the distributions of component values. This assumption will be verified through the use of Monte Carlo simulations in Chapter 3.4.3.3.

3.3.2 Optimization

The average unit cost, U, can be minimized by optimizing manufacturing rates of components and the tolerance spread. As a reminder, for a given process, the production rate affects the processing cost and the precision of the process. This is an unconstrained multivariate optimization problem. The problem can be solved using a variety of optimization methods, including gradient based optimization algorithms (e.g., method of moving asymptotes) [111]. Compared to optimization methods such as Monte Carlo method and Heuristic method, gradient-based methods have advantages such as easy to implement, low storage requirement, and easy to generalize (little effort of parameter tuning). A gradient-based optimization procedure was used in the present research.

The procedure of the gradient-based optimization method is straightforward. We first find the partial derivatives of the average unit cost, U, with respect to all variables (for scenario one, only production rates; for scenario two, both production rates and tolerance spreads). Then the partial derivatives with respect to intermediate variables and approximations for non-elementary functions are needed. The details are given in the sections below.

3.3.2.1 Preparing for Optimization of Scenario One

For scenario one, *U* only depends on the production rates, r_i, and is defined by Equation (3.14). For the gradient based optimization procedure, we must be able to evaluate the partial derivative of *U* with respect to r_i, with σ_i being the intermediate variable. The derivative of *U* with respect to r_i is:

$$\frac{\partial U}{\partial r_i} = \left[\operatorname{erf}\left(z\right)\right]^{-2}\left(\frac{d}{dq}\operatorname{erf}\Big|_{q=z}\right)\frac{z}{\sigma_y}\frac{\partial \sigma_y}{\partial r_i}[S_P + \sum_{j=1}^{m} C_{Bj}] + \left[\operatorname{erf}\left(z\right)\right]^{-1}\frac{d}{dr_i}C_{Bi}\,, \tag{3.24}$$

where z is given by:

$$z = \frac{t_y}{\sqrt{2}\sigma_y}\,. \tag{3.25}$$

As the standard deviation of the product, σ_y, is dependent on the production rates of individual components, the partial derivative of σ_y with respect to r_i is:

$$\frac{\partial \sigma_y}{\partial r_i} = \left(\sum_{j=1}^{m} \delta_j{}^2 \sigma_{xj}{}^2\right)^{-\frac{1}{2}} \delta_i^2 \sigma_{xi}\frac{d\sigma_{xi}}{dr_i}\,. \tag{3.26}$$

With Equations (3.24) to (3.26) in place, the values of r_i may be optimized to minimize *U*.

3.3.2.2 Preparing for Optimization of Scenario Two

For scenario two, *U* depends on both r_i and k_i, and is defined by Equation (3.18). Again, the gradient based optimization procedure requires values for the partial derivative of *U* with respect to r_i and k_i, with σ_i being the intermediate variable. The derivative of *U*, with respect to r_i is:

$$
\begin{aligned}
\frac{\partial U}{\partial r_i} = {} & \left[\operatorname{erf}\left(z\right)\right]^{-2}\left(\frac{d}{dq} erf\left|_{q=z}\right.\right)\frac{z}{\sigma_y}\lambda\frac{\partial \sigma_y}{\partial r_i}\left[S_P+\sum_{j=1}^{m}\left(\frac{C_{Bj}}{\operatorname{erf}\left(\frac{k_j}{\sqrt{2}}\right)}+\frac{S_{Cj}}{\operatorname{erf}\left(\frac{k_j}{\sqrt{2}}\right)}-S_{Cj}\right)\right] \\
& +\left[\operatorname{erf}\left(z\right)\right]^{-1}\left[\operatorname{erf}\left(\frac{k_i}{\sqrt{2}}\right)\right]^{-1}\frac{d}{dr_i}C_{Bi},
\end{aligned}
\tag{3.27}
$$

where, z is:

$$
z=\frac{t_y}{\sqrt{2}\sigma_y}. \tag{3.28}
$$

The partial derivative of σ_y, with respect to r_i is given by:

$$
\frac{\partial \sigma_y}{\partial r_i}=\left(\sum_{j=1}^{m}\left(\delta_j h(k_j)\sigma_{xj}\right)^2\right)^{-\frac{1}{2}}\delta_i^{\,2}\left(h(k_i)\right)^2\sigma_{xi}\frac{\partial \sigma_{xi}}{\partial r_i}, \tag{3.29}
$$

and the derivative of U with respect to k_i is:

$$
\frac{\partial U}{\partial k_i}=\left[\operatorname{erf}\left(z\right)\right]^{-2}\left[\begin{array}{l}
\left(\frac{z}{\sigma_y}\right)\frac{\partial \sigma_y}{\partial k_i}\left(\frac{d}{dq}\operatorname{erf}\left|_{q=z}\right.\right)\left(S_p+\sum_{j=1}^{m}\left(\frac{C_{Bj}+S_{Cj}}{\operatorname{erf}\left(\frac{k_i}{\sqrt{2}}\right)}-Sc_j\right)\right) \\
-erf\left(z\right)\left[\operatorname{erf}\left(\frac{k_i}{\sqrt{2}}\right)\right]^{-2}\frac{S_{Ci}+C_{Bi}}{\sqrt{2}}\left(\frac{d}{dq}\operatorname{erf}\left|_{q=\frac{k_i}{\sqrt{2}}}\right.\right)
\end{array}\right], \tag{3.30}
$$

where,

$$\frac{\partial \sigma_y}{\partial k_i} = \left(\sum_{j=1}^{m} \left(h(k_j)\sigma_j \delta_j \right)^2 \right)^{-\frac{1}{2}} \delta_i^2 \sigma_i^{\ 2} h(k_i) \frac{dh}{dq}\Big|_{q=k_i} . \tag{3.31}$$

3.3.2.3 Approximation of the Gauss Error Function and Its Derivative

The Gauss error function, erf(*x*), is a non-elementary function, its value can be approximated using an nth order Maclaurin series:

$$\operatorname{erf}(x) = S_N(x) + R_N(x) , \tag{3.32}$$

where, $S_N(x)$ is:

$$S_N(x) = \frac{2}{\sqrt{\pi}} \int_0^x \sum_{j=0}^{n} (-1)^n \frac{q^{2j}}{j!} dq = \frac{2}{\sqrt{\pi}} \sum_{j=0}^{n} (-1)^j \frac{x^{2j+1}}{(2j+1)j!} , \tag{3.33}$$

and $R_N(x)$ is the remainder. By using the Leibniz Criterion, the remainder satisfies the following inequality equation:

$$\left| R_N(x) \right| \le \frac{2}{\sqrt{\pi}} \int_0^x \frac{q^{2(n+1)}}{(n+1)!} dq = \frac{2x^{2n+3}}{\sqrt{\pi}(2n+3)(n+1)!} . \tag{3.34}$$

Similarly, the derivative of the Gauss error function is:

$$\frac{d}{dx}\operatorname{erf} = \frac{d}{dx} S_N(x) + \frac{d}{dx} R_N(x) = \frac{2}{\sqrt{\pi}} \sum_{j=0}^{n} (-1)^j \frac{x^{2j}}{j!} + D_N(x) , \tag{3.35}$$

where, $D_N(x)$ is the remainder of the derivative of the Gauss error function, which satisfies the following inequality equation:

$$\left|D_N(x)\right| \leq \frac{2x^{2(n+1)}}{\sqrt{\pi}(n+1)!} \,. \tag{3.36}$$

The upper bound of approximation errors (remainders) R_N and D_N can be estimated by Equation (3.34) and Equation (3.36). The larger the value of n, the smaller the approximation error. For any given maximum allowable error, there exists a lower bound of n that satisfies this requirement.

3.4 Case Study

To validate the proposed method, an overrunning clutch product is considered [112]. This is an example commonly used in the literature on tolerance allocation. Using this example makes it easier to compare different methods. Some traditional optimization methods have been applied in solving the problem, for example, worst case tolerance analysis [19] and brute-force search [12]. Because the simple optimization models could not find optimal/near optimal solutions, these methods tend to generate tight tolerance. Some heuristic based algorithms have also been used, such as particle swarm optimization [22] [113] and genetic algorithm [20], these algorithms usually requires the tuning of optimization parameters, which are difficult to be compared. Thus, the proposed method is compared with a heuristic method from the literature that does not need parameter tuning. The advantages of the proposed methods are analyzed in the comparison.

To assess the accuracy of the proposed statistical approach, as described in Eqs.(3.8)-(3.12), a Monte Carlo simulation is employed. Analyses are carried out to compare the two scenarios (no component inspection and 100% component components).

3.4.1 Problem Description

An overrunning clutch is assembled from three types of components, i.e., hub, roller, and cage. For proper functioning, the contact angle (product characteristic), y, of an overrunning clutch should be within ±0.035 rad from the nominal value of y_0=0.122 rad (t_y =0.035). The value of y for an assemblage of components is given by the following design function:

$$y = f(x_1, x_2, x_3) = \arccos(\frac{x_1 + x_2}{x_3 - x_2}), \tag{3.37}$$

where, x_1 is the diameter of the hub, x_2 is the diameter of the roller, and x_3 is the inner diameter of the cage (please refer to Figure 3.7). The nominal values of x_1, x_2, and x_3 are 55.29 mm, 22.86 mm, and 101.69 mm. The tolerance of the clutch must be allocated to the three components.

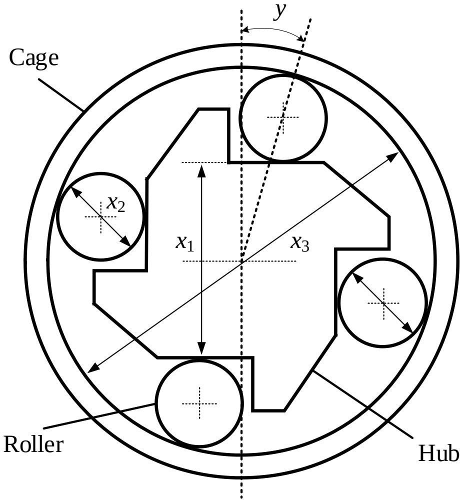


Figure 3.7. An overrunning clutch [112].

3.4.2 Rate-Cost and Rate-Sigma Relationships

In the literature, production rates have been in such terms as material removal rate [18] and units per unit time [102]. In this case study, the production rate, r, is expressed in terms of the number of components produced per minute. For a single component, the processing cost, C_B, is:

$$C_B = A + \frac{B}{r}, \tag{3.38}$$

where, *A* is the fixed cost per part such as set-up cost, and *B/r* is a production rate-dependent cost. The values of *A* for the three components are shown in Table 3.2.

Turning attention to the production rate-dependent term in the processing cost, its coefficient, *B*, consists of the following:

$$B = B_M + B_L + B_E, \tag{3.39}$$

where, B_M is the cost of the machine tool (includes machine depreciation and operating cost), B_L is the cost of labor, and B_E is the cost of electricity. These three coefficients are estimated using typical cost rates for use of machine tools, labor, and electricity (all expressed in \$/min), as shown in Table 3.3. The relation between *r* and C_B is shown in Figure 3.8.

Table 3.2. Values of the fixed cost, *A*.

Component	*A* ($)
Hub	0.98
Roller	0.52
Cage	1.22

Table 3.3. Values of *B*.

Component	B_M ($/min)	B_L ($/min)	B_E ($/min)	*B* ($/min)
Hub	0.142	0.500	1.494	2.136
Roller	0.150	0.500	1.150	1.800
Cage	0.368	0.500	1.700	2.568

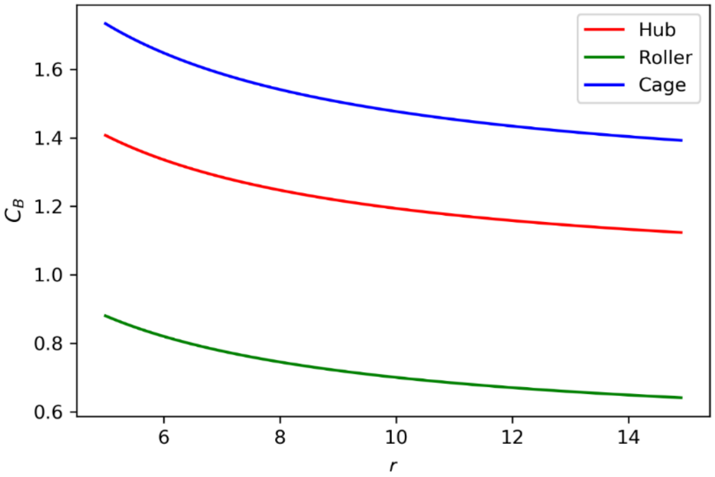


Figure 3.8. The relation between r and C_B.

Let us assume that machining processes are used to produce the components. In machining, feed rate is a major factor that affects production rate. It also affects the precision of the process, as it is related to measures such as cutting forces, which influence deflection [114]. We propose that the precision (σ) is linearly related to the square of feed rate. This hypothesis is supported by several studies from the literature. For example, Yeh and Hsu showed that the tolerance value of a chord error, G, is linearly related to the square of the feed rate of a CNC machine [115]. Boothroyd and Knight demonstrated that the roughness (R_a) of a process is linearly related to the square of feed rate [116]. Lim and Meng expressed the cutting force as a second degree polynomial function of the feed rate [117]. Based on these relations from the literature, the following expression is proposed to describe the value of σ:

$$\sigma_{xi} = E_i + F_i r_i^2 , \tag{3.40}$$

where E_i and F_i are coefficients corresponding to the process used to fabricate the ith component type. E_i provides a lower bound for σ (the best precision the process can achieve), and F_i affects the shape of the curve. Figure 3.9 shows the assumed behavior of σ as a function of r for the given case (adapted from [12]). The values used for E_i and F_i in the current study are given in Table 3.4.

Table 3.4. Values of coefficients for E_i and F_i.

Component	E_i (mm)	F_i (mm·min^2)
Hub	0.0320	4×10^{-4}
Roller	0.0214	2×10^{-4}
Cage	0.0534	6×10^{-4}

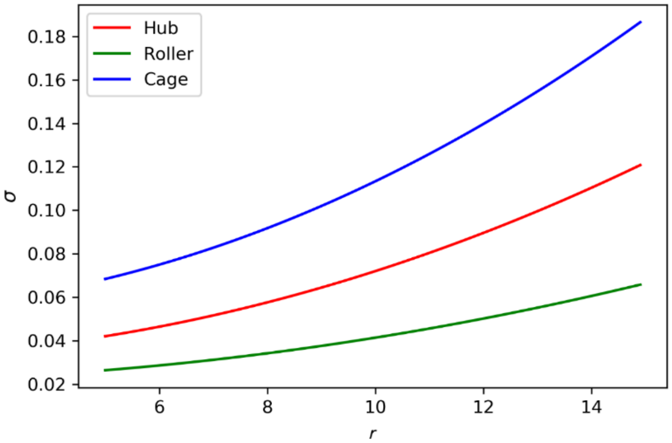


Figure 3.9. The relation between r and σ.

3.4.3 Results and Analysis

For the given design function, Equation (3.37), the derivatives, δ_1, δ_2, and δ_3, may be determined using the general expression of Equation (3.12). As has been noted, these derivatives are needed to compute σ_y in Equation (3.11). For the specified design function (Equation. (3.37)), the derivative relationships are:

$$\delta_1 = \frac{\partial y}{\partial x_1}\Big|_{\mu_1,\mu_2,\ldots,\mu_m} = \left(\frac{-1}{\sqrt{1-\left(\frac{x_1+x_2}{x_3-x_2}\right)^2}}\frac{1}{x_3-x_2}\right)\Big|_{\mu_1,\mu_2,\ldots,\mu_m}, \tag{3.41}$$

$$\delta_2 = \frac{\partial y}{\partial x_2}\Big|_{\mu_1,\mu_2,\ldots,\mu_m} = \Big(\frac{-1}{\sqrt{1-(\frac{x_1+x_2}{x_3-x_2})^2}}\frac{x_1+x_3}{(x_3-x_2)^2}\Big)\Big|_{\mu_1,\mu_2,\ldots,\mu_m}, \tag{3.42}$$

$$\delta_3 = \frac{\partial y}{\partial x_3}\Big|_{\mu_1,\mu_2,\ldots,\mu_m} = \frac{1}{\sqrt{1-(\frac{x_1+x_2}{x_3-x_2})^2}}\frac{x_1+x_2}{(x_3-x_2)^2}\Big|_{\mu_1,\mu_2,\ldots,\mu_m}. \tag{3.43}$$

For the ith component type, to optimize the production rate values, r_i, across all values for i, the derivative of Cp_i and σ_i, with respect to r_i must be determined. Based on the relation between r and Cp, as given by Equation (3.38), and the relation between r and σ, as given by Equation (3.40), the sensitivity of the processing cost, C_B and process variance, σ^2, to changes in the production rate for each process are:

$$\frac{dC_{Bi}}{dr_i} = -B_i r_i^{-2}, \tag{3.44}$$

$$\frac{d\sigma_{xi}}{dr_i} = 2F_i r_i. \tag{3.45}$$

The open source optimization library, NLopt [118], was employed to solve the cost optimization problem.

3.4.3.1 Scenario One: No Inspection of Components

First, scenario one, as described in Section 3.3.1.1, was considered. Recall that this scenario either does not inspect individual components or uses acceptance sampling (and based on the sample, the lot is judged to be satisfactory). The components are assembled to create a product, and then these products are inspected. An additional cost is incurred if a product is deemed to be unsatisfactory. This cost, S_P, is associated with the additional expense of scrapping a product. The value of S_P was assumed to be ten percent of the summation of the fixed costs, A, of all components that are assembled into the product. The average unit cost of a satisfactory product, U, is defined by Equation (3.14). The heuristic algorithm-based tolerance allocation method proposed in Chapter 2 was compared with the method proposed in this chapter. The heuristic method allocated product tolerances by optimizing the σ of the processes. The number of each type of component that was processed and assembled, Q, was set equal to 10,000.

The results of the heuristic method in Chapter 2 and the approach proposed in this chapter are shown in Table 3.5. Using the method proposed in this chapter, a lower average unit cost per satisfactory product is achieved (a \$150 difference for 10,000 parts). Furthermore, 235 more satisfactory products can be produced per 10,000 units of components processed (≈2%).

Table 3.5. Results of scenario one.

	Method of chapter 2	Method of this chapter	
Component	σ	r	σ
Hub	0.077	10.909	0.080
Roller	0.046	10.149	0.042
Cage	0.118	9.284	0.105
U (\$)	3.5002	3.485	
M	9410	9645	

3.4.3.2 Scenario two: 100% inspection of components

This section considers scenario two that was described in Section 3.3.1.2. Recall that this scenario inspects every component prior to assembly. The cost to scrap/recycle unsatisfactory components, S_C, is assumed to be ten percent of the fixed cost, A, of the process used to fabricate the component. As in Section 3.4.3.1, the cost to scrap/recycle unsatisfactory products, S_P, is assumed to be ten percent of the summation of the fixed costs, *A*, of all components that are assembled into the product. The settings of the Monte Carlo simulation are the same as in Section 3.4.3.1. The results using the method of this research are again compared with the heuristic algorithm-based tolerance allocation method Chapter 2. The results of the two methods are shown in Table 3.6.

Compared to the heuristic method proposed in Chapter 2, the method proposed in this chapter achieves a smaller average unit cost per satisfactory product (a $160 difference for 10,000 parts), and assembled 254 more satisfactory products per 10,000 satisfactory products produced (≈2.5%). The two methods have similar component pass rates. In the heuristic method, the production rate (derived from σ) and tolerance spread, k, had to be optimized separately, while in the proposed method of this research, the production rate and tolerance spread were optimized together. Thus, a better solution was reached in terms of cost, quality, and waste reduction (material efficiency) using the method of this research.

Table 3.6. Results of scenario two.

	Method of chapter 2			Method of this chapter			
Component	σ	k	N	r	σ	k	N
Hub	0.077	3.484	10004	10.9	0.080	4.071	10000
Roller	0.046	3.785	10001	10.2	0.042	3.030	10024
Cage	0.118	3.723	10003	9.3	0.105	3.535	10003
U ($)		3.500			3.484		
M		9401			9655		

Several observations associated with the two scenarios, as provided in Tables 4 and 5, merit attention. The results of both methods for both scenarios produce a relatively high percentage of unsatisfactory assembled products (about 3%). Across all cases, the objective of minimizing the average unit cost of a satisfactory product, *U*, is pursued. The obtained solutions are optimal from a cost perspective given the capabilities (precisions) and costs of the processes. These results can help practitioners adjust the production process. For example, if the current scrap rate is unacceptable, then this may indicate that the capabilities of the processes need to be improved or that the scrap cost is not large enough in the model.

3.4.3.3 Further analysis

COMPARISON OF THE TWO SCENARIOS. In an ideal world, it would be desirable for a process to be sufficiently precise so that no inspection of components is needed. However, when the precision of the process is limited, or when it is expensive to increase the precision of the process, inspection may be a cost-effective strategy to ensure that the components/products have acceptable quality. Comparison of the cost for the two scenarios will help practitioners make a decision as to whether to carry out inspection. As is evident from a comparison of the results of Table 3.5 and Table 3.6, the inspection of components, does not dramatically affect either U or M. However, this result may be dependent on the precision (σ) of the processes. For processes with poorer precision (bigger σ), the difference may be larger.

In this section, the lower bound for σ of processes is varied, and the average unit cost per satisfactory product of the two scenarios are compared. For scenario two, two cases are considered. For the first case of scenario two, the value of k is fixed at 3 (i.e., specifications are $\pm 3\sigma$), and for the second case of scenario two, the value of k is optimized.

In the production rate-σ model given by Equation (3.40), the value of E is the lower bound on the standard deviation for the process. The precision of the process was varied by multiplying the value of E given in Table 3.4 by a "precision scaling constant." The value of the precision scaling constant was varied from 0.5 to 3.0 (a value of 0.5 reduces the lower bound on σ by ½ and a value of 3.0 increases the low bound on σ by a factor of 3). The value of F was fixed at the value given in Table 3.4. For both scenarios, the cost-production rate relationship was the same as used previously, and the values of the constants *A* and *B* in Equation (3.38) were the same as in Table 3.2 and Table 3.3. Because the cost to inspect the component is independent of production rate, and the component scrap cost is negligible (very few components are scrapped), the inspection cost and component scrap cost were both assumed to be 0. The minimum values of *U* for scenario one and scenario two (two cases for scenario two) for different precision scaling coefficients are shown in Figure 3.10. The figure shows that for processes with high precision, the difference among the three cases is small, so the economic benefit of inspection is negligible. As the precision of the process decreases (σ increases), the economic benefits of inspection increase. Optimizing the value of *k* can reduce *U* – the average unit cost per satisfactory product.

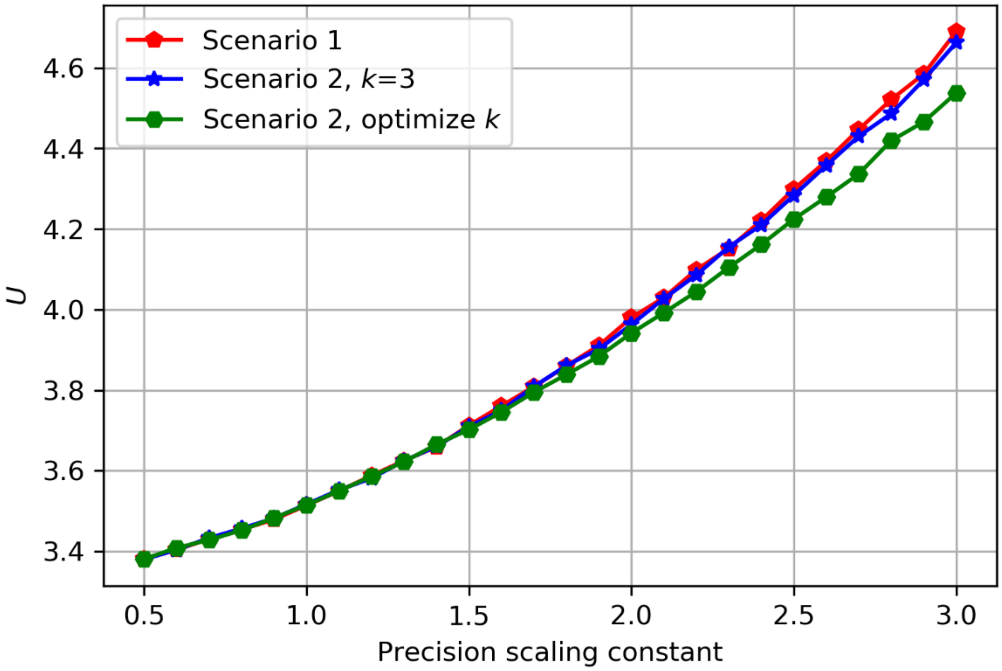


Figure 3.10. Comparison of two scenarios

INFLUENCE OF PRECISION ON TOLERANCE SPREAD AND PASS RATE. This section analyzes the influence of process precision on the allocated tolerance of components. As with the former section, the precision of the process was varied by multiplying the value of E given in Table 3.4 by a precision scaling constant, which was varied from 0.5 to 3.0. The value of F was fixed at the value listed in Table 3.4. For all three cases, the values of the constants A and B in Equation (3.38) are the same as given in Table 3.2-Table 3.3. The values of γ, β, and k for minimum U are shown in Figure 3.11. A general trend for the components is that when the process precision decreases (σ increases), the value of k also decreases, which adjusts the tolerances of the components.

The value of k and the pass rate of the roller are the smallest among the three components, which means the tolerance of the roller is the tightest, even though the process for the roller has the highest precision. Since the process for the roller has the lowest processing cost, tightening the tolerance of the roller reduces the loss caused by assembling expensive hub and cage components that may be scrapped if the product fails inspection due to large variation stack-up. The optimization seems to be telling us that driving down the stack-up variation by focusing on the inexpensive rollers is the least expensive way to affect the average unit product cost, U.

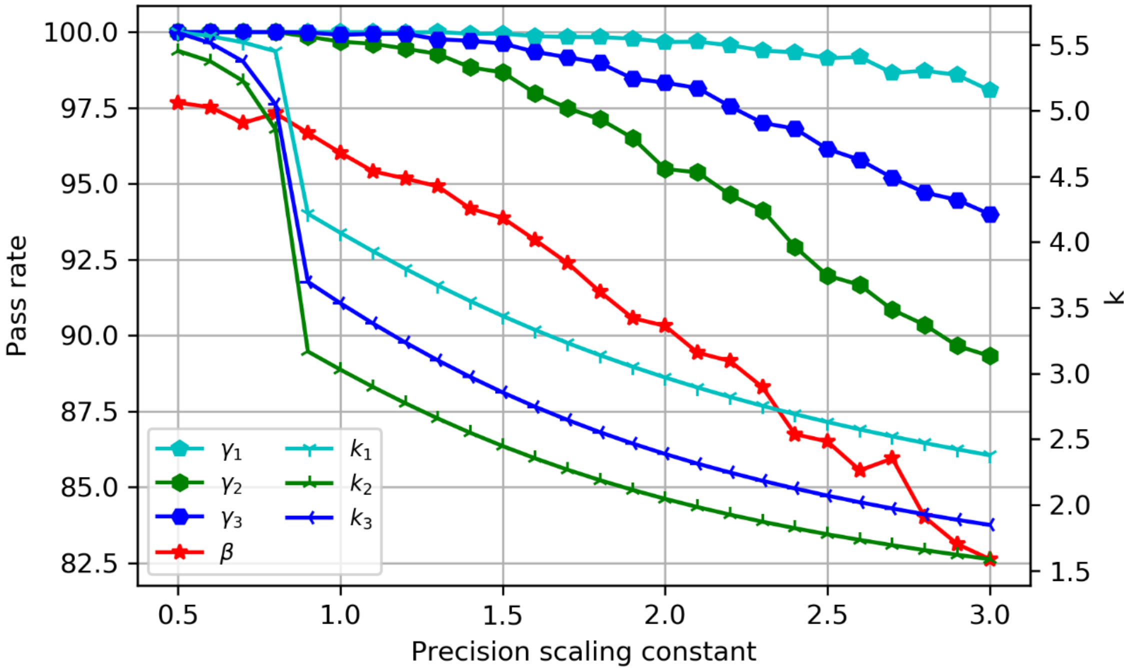


Figure 3.11. Relation among process precision, tolerance spread, and pass rate.

ACCURACY OF THE COST MODEL. In the previous subsections, the optimization results, such as the average unit product cost, *U*, and the number of satisfactory products *M*, were calculated by the proposed analytical model. This section reports on the Monte Carlo simulations that were used to validate the accuracy of the analytical model. Model validation was achieved by comparing the relative error between the average unit product cost estimated by the Monte Carlo simulations and the average unit product cost computed by the proposed cost model (scenario one: Equation (3.14), scenario two: Equation (3.18)).

To review, the proposed optimization method finds the optimal production rates and the minimum average unit cost of a satisfactory product. For the Monte Carlo simulations, the production rates found via the proposed method were used to generate components of each type.

The dimension of each component type was assumed to follow a normal distribution (with mean equal to the nominal value, and standard deviation functionally dependent on the production rate).

Both scenario one and scenario two were considered. For scenario one, 10,000 components of each type were generated. These components were randomly assembled, the number of satisfactory products was counted, and the average unit cost of a satisfactory product was computed. For scenario two, components of each type were again generated, each component was inspected, and unsatisfactory components were scrapped. This process continued until 10,000 satisfactory components of each type were produced. The 10,000 satisfactory components of each type were randomly assembled, the number of satisfactory products was counted, and the average unit product cost was computed.

To validate that the accuracy of the model does not change dramatically with the change of process precision, the Monte Carlo simulations were carried out multiple times with varying precision (multiplying the value of E given in Table 3.3. by a precision scaling constant, which was varied from 0.5 to 3.0). The conditions for the simulations were the same as those given in Section 3.4.3.1 and 3.4.3.2. The relative errors, (Monte Carlo – analytical model)/Monte Carlo, are shown in Figure 3.12. For both cases, the relative errors were all positive, which means the cost predicted by Monte Carlo simulations was larger than the theoretical model. The error being all positive means the product variance estimated from component variances is slightly smaller than the real value. The relative errors in both cases are modest (average of about 1%) and may be compensated by slightly increase the estimated product variance in equation (3.11).

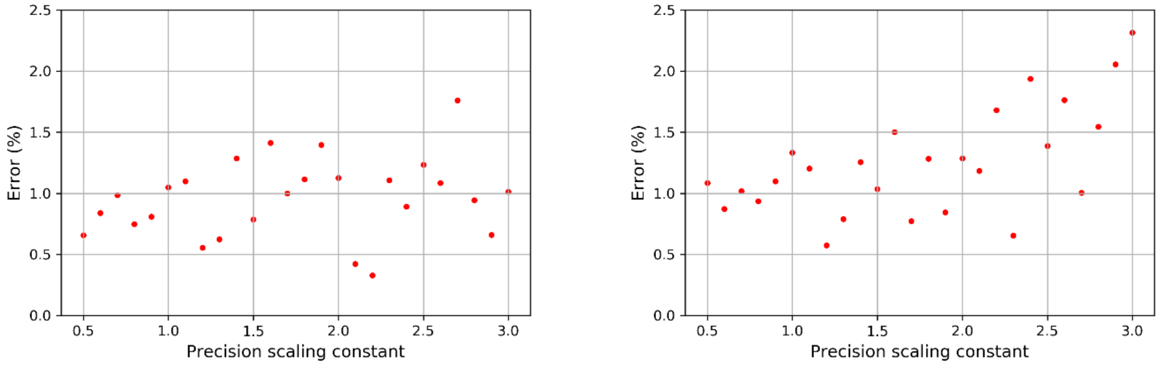


Figure 3.12. Errors between Monte Carlo simulations and analytical model: scenario one (left) and scenario two (right)

MANAGERIAL INSIGHTS. The case study has shown the advantages of jointly considering product design and operation design in allocating tolerance. Based on the analysis from the above three subsections, some managerial insights on inspection strategy, cost reduction, and waste reduction are provided.

The results in Table 3.6 shows that tightening the tolerance of components that is less expensive to process (inexpensive components) and loosening the tolerance of components that is more expensive to process (expensive components) is an effective way to reduce the total cost. Though a tighter tolerance may initially look to increase the cost, this strategy reduces the loss caused by assembling expensive components into a product that fails inspection. Especially when the failing is caused by large variation stack-up from the variation of inexpensive components. It should also be noted that this strategy in saving cost should not be exploited because unnecessary tight tolerance may cause unnecessary precise process to be used, which increases the energy consumption (low production rate and long production time) and increases waste (more components are scrapped).

Quality inspection is generally time consuming and costly. Practitioners must make decisions on whether to carry out inspection based on the precision of the process and the cost of inspection. The comparison in the former subchapter only considered the costs that are directly affected by tolerance and production rate, so the result reveals how the precision of the process affect the cost. The comparison gives practitioners some guidance on the level of process precision at which inspection of components becomes economically desirable. Practitioners should evaluate cost related to inspection, such as investment on equipment and labor, to make decisions.

Costs to scrap products and component are generally low [119]. If the solution to tolerance allocation is driven only by high quality and low cost, a manufacturer may take a strategy that uses low precision processes to produce the components, which would keep manufacturing costs low. Then, a tight tolerance could be applied to the components and products to filter out poor quality assemblies. This strategy would lead to a higher waste. If the manufacturer would like to reduce waste in addition to cost, they can add a "punish" factor into the cost model. The "punish" factor can be represented by the ratio between product scrap cost S_P, and fixed cost, A (or similarly, the ratio between component scrap cost, S_C, and A). A high ratio has a higher punish on waste. We compared three cases, with three ratios being 10%, 50%, and 100%. The results are shown in Table 3.7. It is evident that when the punish factor is increased, less products and components go into the waste stream. Practitioners can adjust this factor based on requirement.

Table 3.7. Impact of scrap cost on cost and waste.

$\frac{S_P}{A}$	N			U ($)	M
	Hub	Roller	Cage		
10%	10000	10000	10000	3.484	9655
50%	10024	10009	10000	3.514	9701
100%	10003	10000	10000	3.544	9746

3.5 Summary and Conclusions

This chapter introduced a new tolerance allocation model that considered production cost, quality, and waste simultaneously. This model for the first time, jointly considers product design and operation design. For product design, a statistical approach was used to predict how component variations contributed to the variation of an assembled product. For operation design, the relations among production rate, processing precision, processing cost, and waste were characterized.

An analytical cost model was proposed. The cost model considered processing cost and scrap cost. Two scenarios were studied: i) no component inspection, and ii) 100% inspection of components (assembled products were always inspected). The tolerance on the product characteristic of interest was allocated to individual components by optimizing the production rate for each component (the production rate affects the processing cost and precision). For the scenario where components were inspected, the tolerance spread, k (tolerances are $\pm k\sigma$), was also optimized. Since component inspection may change the distribution of characteristic values, an adaptation function was introduced to appropriately adjust the standard deviation of components. A gradient-based optimization method was used to minimize the cost. Tolerance allocation for a clutch assembly was used to demonstrate the proposed method. The obtained results were compared with the results of a heuristic-based method. It is shown that the proposed method leads to settings (production rate and tolerance spread) that produce more satisfactory products at a lower cost and produce less waste. In addition, model validation was conducted by comparing the relative error between the average cost computed from Monte Carlo simulations and the average cost computed by the proposed theoretical cost model.

Some conclusions and practical guidance may be drawn from this work as follows.

- Monte Carlo simulations demonstrated the accuracy of the analytical cost model (average of about 1% error rate).

- The case study showed that the proposed method can optimize tolerance by balancing cost, quality, and waste. Compared to a heuristic method from the literature, the proposed method produces more satisfactory products at a lower average unit product cost and lower waste (fewer scrapped/recycled components/products).
- When the precision of a process is high, it is more economical not to inspect the quality of individual components. For poor precision processes (large σ), inspecting the quality of individual components is the preferred approach from cost/throughput standpoint. In the long term, continuous quality improvement should be pursued to improve process precision and minimize waste.
- The cost model and the optimization method were developed based on general functional forms for the rate-cost and rate-sigma relationships. Two specific forms of these relationships were considered in the case study. Manufacturers can establish these relationships that were suitable to their situation of interest.

The ability to produce high quality products with low cost and high throughput rate is critical for manufacturers. This research has demonstrated how allocating product tolerance by jointly considering product design and operation design can help improve this ability while minimizing scrap. In moving forward, it should be noted that continuous improvement approaches (a topic not addressed in this chapter) that reduce process variation while also reducing costs are likely to add to the benefits of this research and produce even smaller levels of scrap/waste.

4. DETECTING ANOMALIES IN TIME SERIES DATA FROM A MANUFACTURING SYSTEM USING RECURRENT NEURAL NETWORKS[4]

The industrial internet of things allows manufacturers to acquire large amounts of data. This opportunity, assuming the right methods are available, allows manufacturers to find anomalies that arise during manufacturing system operation. Data acquired from a manufacturing system are usually in the forms of time series. This chapter proposes a new method that can detect anomalies in time series data. This model is based on recurrent neural networks, and it can be trained using data acquired during routine system operation. This is very beneficial because often, there are few data labeled as anomalies, since anomalies are hopefully rare events in a well-managed manufacturing system. The model takes time series data as an input and reconstructs the input data. Time series data with an anomaly would causes patterns in the reconstruction errors that are inconsistent with error patterns of anomaly-free data. The performance of the proposed method is assessed using data from a diesel engine assembly process. Three common types of anomalies are detected from the time series data. It is shown that the method not only can detect anomalies, but it can also provide insights into the timestep at which the anomaly occurred. This feature helps a manufacturer pinpoint the source of the problem.

4.1 Introduction

The ability to detect anomalies in data collected during the operation of a manufacturing system provides many benefits to manufacturers. If the root cause of a detected anomaly can be

---

[4] Reprinted (portions enhanced/adapted) from Y. Wang, M. Perry, D. Whitlock, and J. W. Sutherland, “Detecting anomalies in time series data from a manufacturing system using recurrent neural networks,” J. Manuf. Syst., December, 2020. https://doi.org/10.1016/j.jmsy.2020.12.007. Published by Elsevier Ltd.

identified, it may reveal potentially useful information about the manufacturing system, such as the health condition of tools [120], the remaining useful life of machinery [121], and the quality of parts/products [122]. Detecting these anomalies enables manufacturers to catch failures and defects early, and to take remedial actions and precautions.

The advent of low-cost sensors, wireless communication, and advances in computing, which collectively are often conveyed as the industrial internet of things, has made available large amounts of data to manufacturers [36]–[38]. This data, from multiple sensors, is collected over time and forms a multidimensional time series. The accumulation of large amounts of time series data brings opportunities to better monitor the manufacturing system, but the anomaly detection process may also become more challenging, because the sensor data may be of longer length and higher dimension. For example, every time step of the time series collected from a motor may be a vector containing acceleration signals of multiple axes and vibration signals of multiple directions for a long period of time [39] [40]. Traditional anomaly detection methods may not be capable of detecting anomalies in time series data of long length and high dimensionality [41].

When things go wrong during the operation of a manufacturing system, anomalous behavior may appear in sensor data collected from the system. Anomalies in the data can be in different forms, such as extreme values. In general, anomalous data show patterns that are different from anomaly-free data (i.e., data collected during routine functioning of the system). For example, as shown in Figure 4.1, in assembling a crank, the rotational angle of the driving equipment and the corresponding torque applied to rotate the crank can be recorded as a time series. Anomalies in the time series may vary in form and reveal quality defects such as improper engagement of gears (chatter shown in abnormal time series 1) and missing components (low torque values shown in abnormal time series 2).

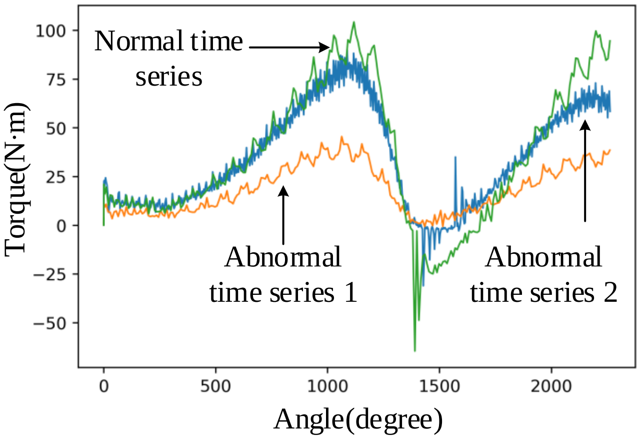
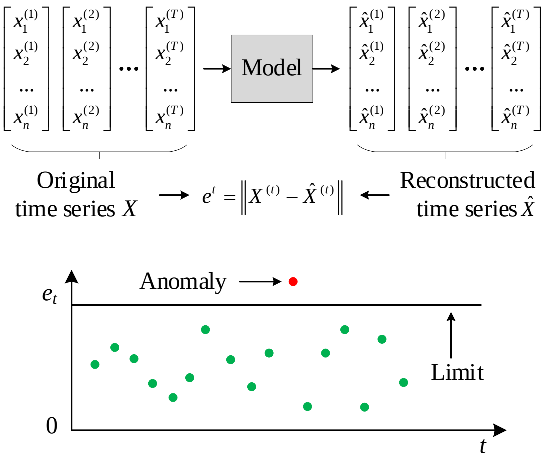


Figure 4.1. Anomalies in time series data.

In this chapter, we use manufacturing (time series) data collected under routine operation to establish a time series model (including estimates of parameters). The differences between the raw data and model predictions serves as the residuals, and the estimation procedure is focused on minimizing these residuals. It is worth noting that different terminologies have been used in the literature. In this research, we refer to the raw data as input data, the model predictions as reconstructed data, and the residuals as reconstruction errors. Since the model was trained using data having no anomalies, the reconstruction errors corresponding to data with anomalies would be large [123]. As shown in Figure 4.2, a time series with a total timesteps of T is noted as $\boldsymbol{X} = \{\boldsymbol{X}_t, t=1,2, \ldots, T, \boldsymbol{X}_t \in \mathbb{R}^n\}$. The reconstruction model takes the original time series data $\boldsymbol{X}$ as input, and outputs a time series data, $\widehat{\boldsymbol{X}} = \{\widehat{\boldsymbol{X}}_t, t = 1,2, \ldots, T, \boldsymbol{X}_t \in \mathbb{R}^n\}$, which is the reconstructed $\boldsymbol{X}$. The reconstruction errors, which are the differences between $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$, are elements of $\boldsymbol{e} = \{\|\boldsymbol{X}_t - \widehat{\boldsymbol{X}}_t\|, t = 1, \ldots, T\}$. The Euclidean norm of the reconstruction errors of each time step can be used as the criteria for detecting an anomaly. By doing so, the problem of detecting anomalies within a time series of multidimensional data is transformed into a problem of detecting anomalies of a series of real numbers.



Model
Original time series X
$e^t = \|X^{(t)} - \hat{X}^{(t)}\|$
Reconstructed time series $\hat{X}$
Anomaly
Limit
e_t
0
t

Figure 4.2. Anomaly detection by analyzing reconstruction errors.

Because of their capability in retaining information of sequential data, recurrent neural networks (RNN) are one of the most commonly used approaches to create such a reconstruction model. The reconstruction process in an RNN model is carried out recurrently; thus, prediction errors of former timesteps accumulate through time. This may limit the effectiveness of using the reconstruction errors as an anomaly detector. In this chapter, we propose an encoder-double decoder model that uses an "attention technique" to solve this limitation. The attention model was proposed by Bahdanau et al. [124] for machine translation based on the encoder-decoder model. We adjusted the attention technique for anomaly detection. By using the attention technique, the encoder better summarizes the content in the input time series by automatically detecting which parts of the input data are more relevant at each time step. The two decoders that share one encoder pass information in the time series both forward and backward through timesteps, which attenuates the accumulation of reconstruction errors.

Another limitation of the RNN based encoder-decoder model is that it may have difficulties characterizing dynamics of high frequency in the input data. In general, there is a tradeoff between time series model complexity and the character of model reconstruction errors (residuals). A simple model may not adequately describe high frequency dynamics in the raw data, or it may well describe high frequency content, but fail to describe a slow drift in the signal. Increasing the complexity of the model allows more features of the raw data to be described, but this comes at the expense of more parameters to estimate. For common RNNs, it is not uncommon for high frequency, low amplitude components in the raw data to appear in the reconstruction errors, even though these components may not be anomalies [61]. We solve this issue by not only monitoring individual reconstruction errors, but also monitoring the trends that are shown in a series. By doing so, more types of anomalies may be detected while avoiding false alarms caused by sharp changes in the time series values.

4.2 Model Description

4.2.1 RNN and the Encoder-Decoder Model

In this section, we describe how to use RNNs to reconstruct a time series of sensor data collected from a manufacturing system and then use the reconstruction error as criteria for detecting anomalies. Generally, an RNN is a neural network that recurrently processes sequential data. At time step t, the RNN maps a sequence of data $\boldsymbol{X}=\{X_1,\dots,X_t\}$ to a fixed length vector h_t, which is a hidden state of the neural network. The hidden state h_t summarizes information in the sequence of data till time step t. Two important factors of an RNN are recurrent computation and parameter sharing, the advantages of which can be explained by a mathematical representation that defines the hidden state, h_t, of an RNN [125]:

$$h_t = g^{(t)}(X_1,\dots,X_t) = f(h_{t-1}, X_t;\theta), \tag{4.1}$$

where, $g^{(t)}$ is a function that takes the sequence of data $\boldsymbol{X}_t$ as an input and maps it to the hidden state h_t. f is a transition function that is shared and applied recurrently at each time step t, and θ is a set of fixed parameters of the transition function *f*. Recurrently applying function *f* means iteratively passing information in the sequence of data through time steps. This iterative computation makes the transition function independent of the number of time steps in the input sequence. And, by sharing the constant θ, only one transition function, *f*, is used at all time steps, rather than using a different $g^{(t)}$ at every timestep. The two factors make an RNN model more generalizable: it is possible to use RNN models for sequences of data with varied lengths and to train the RNN model with fewer training examples [125]. The most common RNN units used to model the function *f* are long short-term memories (LSTM) [126] and gated recurrent unit (GRU) [127].

Since the hidden vector, h_t, summarizes the information of the sequence till time step *t*, a mapping can be built to predict the probability of the sequence data of the next time step being $\hat{X}_{t+1}$ given h_t:

$$p(\hat{X}_{t+1} \mid X_1, \ldots, X_t) = p(\hat{X}_{t+1} \mid h_t). \tag{4.2}$$

By repeatedly carrying out this prediction process, a sequence of data, $\hat{\boldsymbol{X}} = \{\hat{X}_1, \ldots, \hat{X}_T\}$, are reconstructed for the original times series data $\boldsymbol{X}=\{X_1, \ldots, X_t\}$.

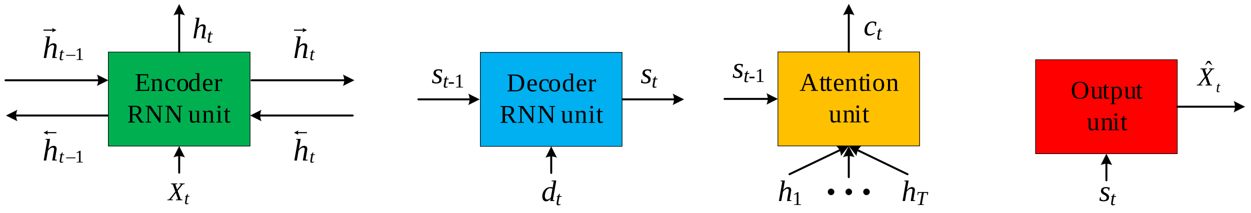
A more powerful RNN model that uses the hidden state to reconstruct the input sequence is the encoder-decoder model [60]. The encoder, which is a series of RNN units, reads the input sequence and then outputs a vector (the hidden state at the last timestep) that summarizes all the information in the time series data. This output vector is passed to the decoder, which is another series of RNN units. Based on the information summarized in the output of the encoder, the decoder reconstructs the original input sequence, $\hat{\boldsymbol{X}} = \{\hat{X}_1, \ldots, \hat{X}_T\}$.

In the encoder-decoder model, information at all timesteps in the input time series is compressed into a fixed length vector. Some useful information in the input time series may be lost. This is the case especially for data series with a long length, where it is difficult for the decoder to find which parts of the sequence are more relevant to reconstructing the data. Bahdanau et al. [124] modified the encoder-decoder model to include the attention model, which is used for machine translation. At each timestep, for each word, the attention model searches the entire input sequence data (a sentence) for the most relevant information to translate the word. By doing so, the output vector of the encoder can place weights on different words based on their importance to predicting the next word. For example, whether a translated verb should be plural may be determined by a noun in the original sentence. An attention model that is adjusted for detecting anomalies within a time series data from a manufacturing system is described in the following section.

4.2.2 The Encoder-Decoder Model with Attention

4.2.2.1 Architecture of the Attention Model

The encoder-decoder model with attention is constructed using four types of neural network units, i.e., the encoder RNN unit, the decoder RNN unit, the attention unit, and the output unit, as shown in Figure 4.3. The encoder unit and the decoder unit consist of a series of general forms of RNN models such as LSTM and GRU. The attention unit and the output unit contain feed forward neural networks. A series of the four types of units are structured together in such a way that the information in the input time series are passed through timesteps. The encoder RNN units take the sequence data as the input, and outputs a sequence of vectors that summarize the information in the input data. For each timestep, the attention units assign weights to these vectors based on their relevance in reconstructing the data of the next timestep. The outputs of the attention units are passed into the decoder RNN units. Then, the output units use the information summarized in the decoder units to reconstruct the input data. Each unit has some parameters to be trained, and for the same types of units, the parameters are shared at all timesteps.



(a) (b) (c) (d)

Figure 4.3. Units of the attention model. (a) Encoder RNN unit. (b) Decoder RNN unit. (c) Attention unit. (d) Output unit.

As shown in Figure 4.3 (a), the encoder RNN unit is a bidirectional RNN, which combines a forward encoder and a backward encoder. The forward encoder reads the input time series data from X_1 to X_T, and outputs the forward hidden states $\vec{h}=\{\overrightarrow{h_t}, t = 1,2 \ldots, T, \overrightarrow{h_t} \in \mathbb{R}^m\}$, which was given by Equation (4.1); the backward encoder reads the input time series data from X_T to X_1, and outputs the backward hidden states $\overleftarrow{h} = \{\overleftarrow{h_t}, t = 1,2 \ldots, T, \overleftarrow{h_t} \in \mathbb{R}^m\}$. The forward hidden states and the backward hidden states are concatenated into a new series of hidden states (h_1, …, h_T), as given below:

$$h_t = \begin{bmatrix} \vec{h}_t \\ \overleftarrow{h}_t \end{bmatrix}. \tag{4.3}$$

By concatenating the hidden states of the forward encoder and the backward encoder, the information of the whole sequence is summarized in the hidden state, h_t, for all time steps from 1 to T, as shown in Figure 4.4. The relevance of the ith hidden state, h_i, in predicting the tth timestep of the output sequence, $\hat{X}_t$, is evaluated by the attention unit. The initial hidden states for the forward RNN unit, $\vec{h}_0$, and backward RNN units of the encoder, $\overleftarrow{h}_T$, are initialized to be zero.

The decoder RNN unit is a unidirectional RNN, as shown in Figure 4.3 (b). The hidden state of the RNN, s_t, is computed recurrently from the hidden state of the former timestep, $s_{i\text{-}1}$, and a vector d_t. The vector d_t is a concatenation of the output of the attention unit (context vector) c_i, and the former reconstructed data $\hat{X}_t$, as shown in Figure 4.4. The initial reconstructed data $\hat{X}_0$ is set to be equal to a random vector or a zero vector. The context vector, c_i, contains both the forward and backward information from the encoder. The hidden state $s_{i\text{-}1}$ and the data $\hat{X}_t$ contain data that have been constructed, which passes the information forward. Using the abstract form of the RNN unit as given in Equation (4.1), the recurrent computation in the decoder can be represented as:

$$s_i = f(s_{i-1}, c_i, \hat{X}_i) = f(s_{i-1}, d_i)\,, \tag{4.4}$$

where, d_i is the concatenation of c_i and X_i, and is given below:

$$d_i = \begin{bmatrix} c_i \\ \hat{X}_i \end{bmatrix}. \tag{4.5}$$

As shown in Figure 4.3 (c), at time step t, the inputs to the attention unit are hidden states of the encoder for all timesteps, $\{h_1, \ldots, h_T\}$, and the hidden state of the decoder of the previous timestep, $s_{t\text{-}1}$. The attention unit learns the relevance of these hidden states and outputs a context vector c_t, which is a weighted sum of all the hidden states. The architecture of the hidden states is given in Sec. 4.2.2.2. Both the decoder RNN unit and the attention unit need an initial hidden state, s_0, to start the recurrent computations. The value of s_0 is calculated by the last hidden state of the backward RNN unit of the encoder, $\overleftarrow{h_1}$, using a one-layer feedforward neural network

The output unit is a feedforward neural network that takes the output of the decoder unit, s_t, as the input and reconstructs the data of step t, $\hat{X}_t$, as shown in Figure 4.3 (d). Based on the complexity of the problem, the structure of the neural network can be adjusted. Also, based on the type of the sequence data X, different activation functions can be deployed at the last layer. For example, if X_i is a vector that indicates which category this time step belongs to, then each element of X_t may be modeled as the probability corresponding to the category, and the softmax function may be deployed as the activation function of the last layer. If X_i is a vector of real values, then a linear transformation may be the activation function of the last layer.

Figure 4.4 shows the structure of the encoder-decoder model with attention. Similar to the encoder-decoder model, the attention model structure can also be divided into an encoder and a decoder. The encoder includes the encoder RNN units and the attention units. The decoder includes the decoder RNN units and the output units. The information in the time series data, $\{X_1, \dots X_T\}$, is transformed from bottom-up, i.e., from the encoder to the decoder. But different from the normal encoder-decoder model, both the encoder RNN unit and the attention unit process all the information at each timestep. The information is passed from the encoder to the decoder through the attention unit. For the decoder RNN unit and the output unit, the information is passed from the past to the future (left to right). A vector $\hat{X}_0$, which can be a random vector or a zero vector, is passed to the first decoder RNN unit to start the decoding process. After carrying out this recurrent computation T times, a sequence of data, $\{\hat{X}_1, \dots, \hat{X}_T\}$, is generated.

The four units of the attention model can be viewed as four function approximators. The four units learn four functions that pass the information from the input time series data, $\{X_1, \dots X_T\}$, to hidden vectors, context vectors, and then to the reconstructed time series data, $\{\hat{X}_1, \dots, \hat{X}_T\}$), as given in Equation (4.6):

$$\begin{aligned} h_t &= \theta(X_1,\ldots,X_T) \\ c_t &= \tau(s_{t-1},h_1,\ldots,h_T) \\ s_t &= \lambda(X_{t-1},s_{t-1},c_t) \\ \hat{X}_t &= \gamma(X_1,\ldots,X_T) = \varsigma(s_t) \end{aligned} \quad , \tag{4.6}$$

where, θ, τ, λ, and γ are functions associated with the encoder RNN unit, the attention unit, the decoder RNN unit, and the output unit, respectively. By sharing parameters, only one function is learned for each type of unit.

In the training process, the objective function, $L = \frac{1}{T}\sum_{t=1}^{T}\left\|X_t - \hat{X}_t\right\|^2$ is minimized. The objective function is the mean of the square of the l_2-norm of the difference vector between the input time series data and the reconstructed time series data.

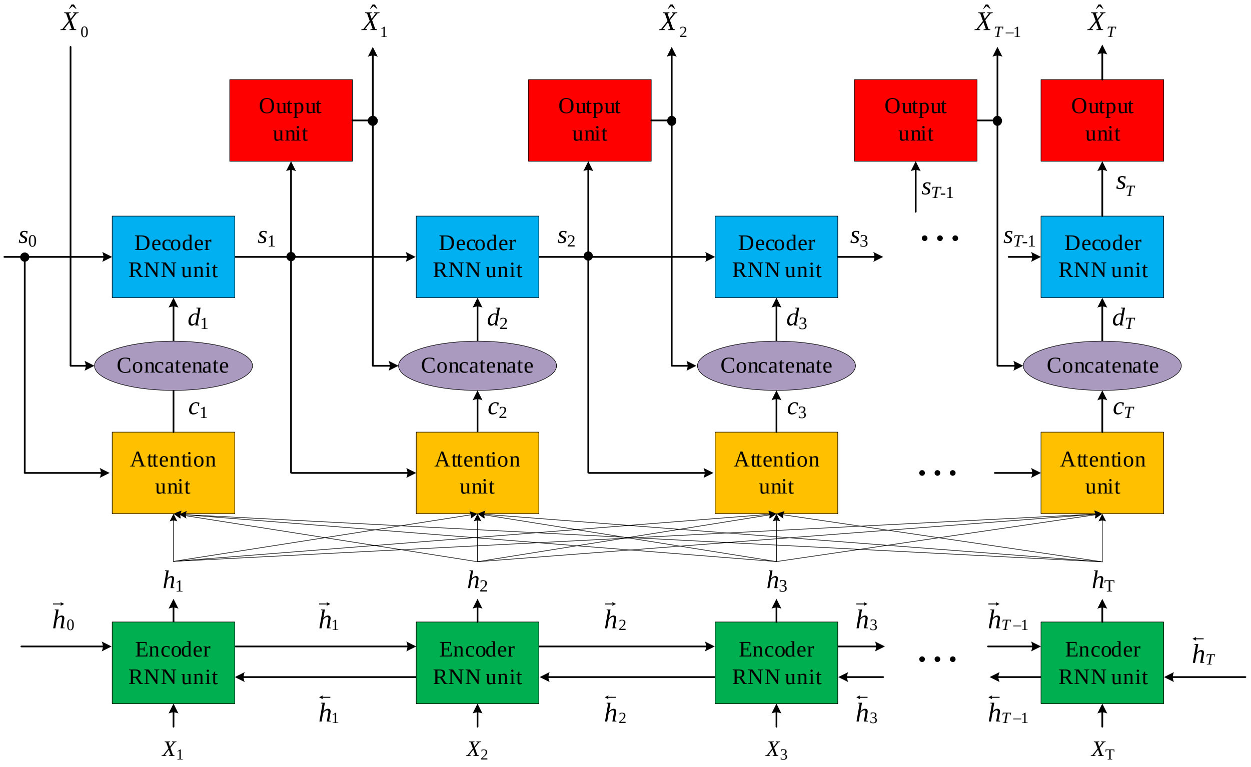


Figure 4.4. RNN Encoder-Decoder architecture.

4.2.2.2 Architecture of the attention unit

The attention unit is called at every timestep by the encoder, and it searches the input sequence data to find the most relevant information that the decoder should use to reconstruct the data for the next time step. The relevance of the data at a time step is evaluated by a weight, and a weighted sum of all the time steps is the context vector c_t.

The architecture of the attention unit is shown in Figure 4.5. At timestep t, the output of the encoder, $\{h_1, \ldots, h_T\}$, and the hidden state of the decoder RNN unit of the previous timestep, s_{t-1}, are the inputs to the attention unit. The context vector c_t is given by the following equation:

$$c_t = \sum_{i=1}^{T} \alpha_{ti} h_i \ , \tag{4.7}$$

where α_{ti} is the weight that evaluates the importance/relevance of the input X_i to reconstruct $\hat{X}_t$. Bahdanau et al [124] proposed the idea of using a feed forward neural network to learn the function that approximates the weight α_{ti}. The value of α_{ti} is between 0 and 1, thus, a softmax function is deployed as the activation function of the output layer. To reconstruct a sequence with T timesteps, the attention network is called T^2 times, the high order of computational complexity limits the complexity of the attention network. The attention network is trained together with the encoder and decoder.

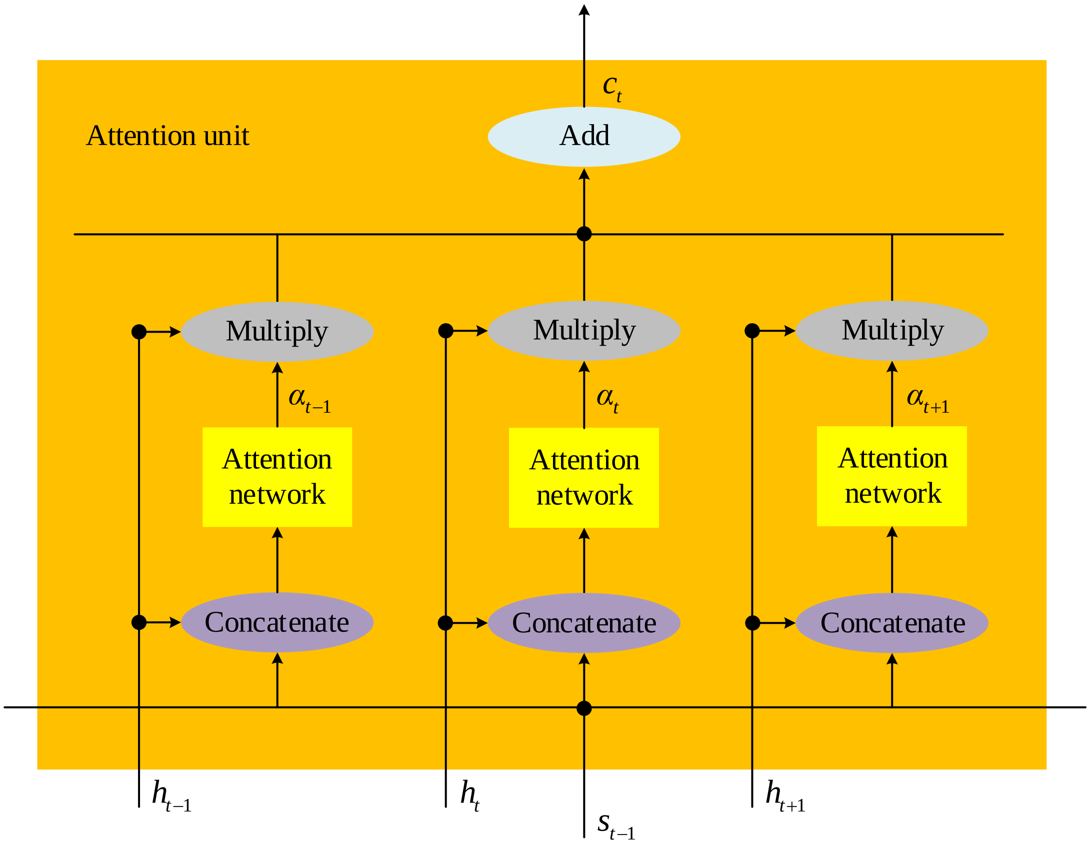


Figure 4.5. The attention unit.

4.2.3 The Encoder-Double Decoder Model

Unlike the encoder, which passes information both forward and backward through timesteps, the decoder in the architecture shown in Figure 2.4, only passes information forward. At timestep t, the decoder reconstructs $\hat{X}_t$ based on s_t and $\hat{X}_{t-1}$, which only reflects the information from earlier timesteps. Thus, the prediction error of $\hat{X}_1, \ldots \hat{X}_{t-1}$, will accumulate through timesteps. For a long series of data, the accumulated error will be mixed with the errors caused by anomalies from the input timeseries. The accumulation of reconstruction errors makes the anomaly detection task more difficult.

We propose an encoder-double decoder model that helps avoid the problems caused by the accumulated reconstruction errors. As shown in Figure 4.6, two decoders, i.e., a forward decoder and a backward decoder, are used. The RNN units of the two decoders have the same structure but separate sets of parameters. A series of forward decoder RNN units are connected and pass information from timestep 1 forward to *T*, and a series of backward decoder RNN units are connected and pass information from timestep *T* backward to 1.

To limit the number of parameters, so that fewer samples are needed to train the model, some of the parameters of the units are shared. For the forward and backward reconstruction processes, only one set of parameters are used for the encoder RNN units, the attention units, and the output units, respectively.

As shown in Figure 4.6, when the attention unit generates the context vector, $\vec{c}_t$, for the forward decoder RNN unit, it takes the hidden states of the forward decoder RNN unit of the former step, $\vec{s}_{t-1}$, and the outputs of the encoder, {h_1, …, h_T}, as inputs. Similarly, when the attention unit generates attention for the backward decoder RNN unit, $\overleftarrow{c}_t$, it takes the hidden states of the backward decoder RNN unit of the former step, $\overleftarrow{s}_{t+1}$, and the outputs of the encoder, {h_1, …, h_T}, as inputs. The forward context vector $\vec{c}_t$ is concatenated with the data reconstructed by the forward decoder RNN unit, $\hat{\vec{X}}_{t-1}$. The concatenated vector is fed into the forward decoder RNN unit to generate $\vec{s}_t$. The vector $\vec{s}_t$ passes information forward and is fed into the output unit to reconstruct $\hat{\vec{X}}_t$. And the backward context vector, $\overleftarrow{c}_t$, is concatenated with the data reconstructed by the backward decoder RNN unit, $\hat{\overleftarrow{X}}_{t+1}$. The concatenated vector is fed into the backward decoder RNN unit to generate $\overleftarrow{s}_t$. The vector $\overleftarrow{s}_t$ passes information backwards, and it is fed into the output unit to reconstruct $\hat{\overleftarrow{X}}_t$. By carrying out this process for *T* timesteps both for the forward reconstruction process and the backward reconstruction process, two series of data are reconstructed, i.e., data reconstructed forward, $\{\hat{\vec{X}}_1, \dots, \hat{\vec{X}}_T\}$, and data reconstructed backward, $\{\hat{\overleftarrow{X}}_1, \dots, \hat{\overleftarrow{X}}_T\}$.

Similar to the case with one decoder, the initial hidden state of the forward decoder RNN unit and the first input to the attention unit, $\vec{s}_0$, are calculated by the last hidden state, $\overleftarrow{h}_1$, of the backward RNN unit of the encoder, using a one-layer linear neural network. Note $\overleftarrow{s}_{T+1}$ as the initial hidden state of the backward decoder RNN unit and the first input to the attention unit that connects to the backward decoder, its value is calculated by the last hidden state, $\vec{h}_T$, of the forward RNN unit of the encoder, using another one-layer linear neural network. The initial reconstructed data for the forward decoder RNN unit, $\hat{\vec{X}}_0$, and the backward decoder RNN unit, $\hat{\overleftarrow{X}}_{T+1}$, are zero vectors.

Since the parameters of the encoder RNN unit, the attention unit, and the output unit are shared by the forward decoder and the backward decoder, the training process should be synchronized in the forward and backward reconstruction processes. This can be achieved by combining the objective functions of the two decoders, as shown in Equation (4.8):

$$L = \vec{L} + \overleftarrow{L} = \frac{1}{T}\sum_{t=1}^{T}\left(\left\|\vec{e}_t\right\|^2 + \left\|\overleftarrow{e}_t\right\|^2\right) = \frac{1}{T}\sum_{t=1}^{T}\left(\left\|X_t - \overrightarrow{X}_t\right\|^2 + \left\|X_t - \overleftarrow{X}_t\right\|^2\right), \tag{4.8}$$

where, $\vec{\boldsymbol{e}} = \{\vec{e}_t = X_t - \hat{\vec{X}}_t, t = 1, \dots, T\}$, and $\overleftarrow{\boldsymbol{e}} = \{\overleftarrow{e}_t = X_t - \hat{\overleftarrow{X}}_t, t = 1, \dots, T\}$ are reconstruction errors corresponding to the forward decoder and the backward decoder.

The training is carried out by back-propagation. The neural networks shared by the two decoders automatically learn to generalize information for the forward and backward reconstruction processes, and the two independent neural networks for the forward decoder and the backward decoder learn independently.

At each timestep, the reconstruction error vector with the smaller norm is saved in the reconstruction error vector, $\boldsymbol{e} = \{\min(\|\vec{e}_t\|, \|\overleftarrow{e}_t\|), t = 1, \dots, T\}$. This has two benefits: 1) avoiding accumulation of reconstruction errors. 2) removing patterns brought in by noisy outliers.

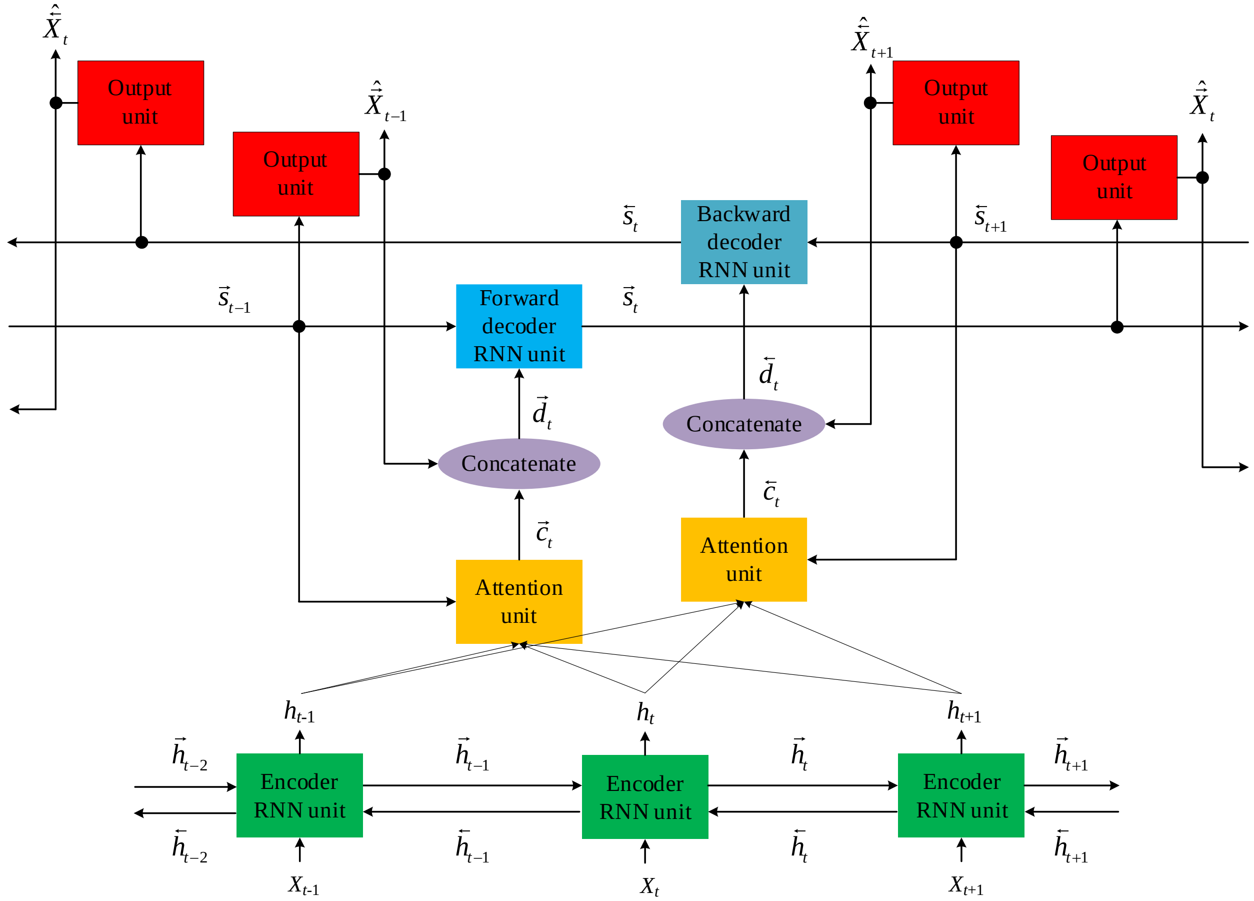
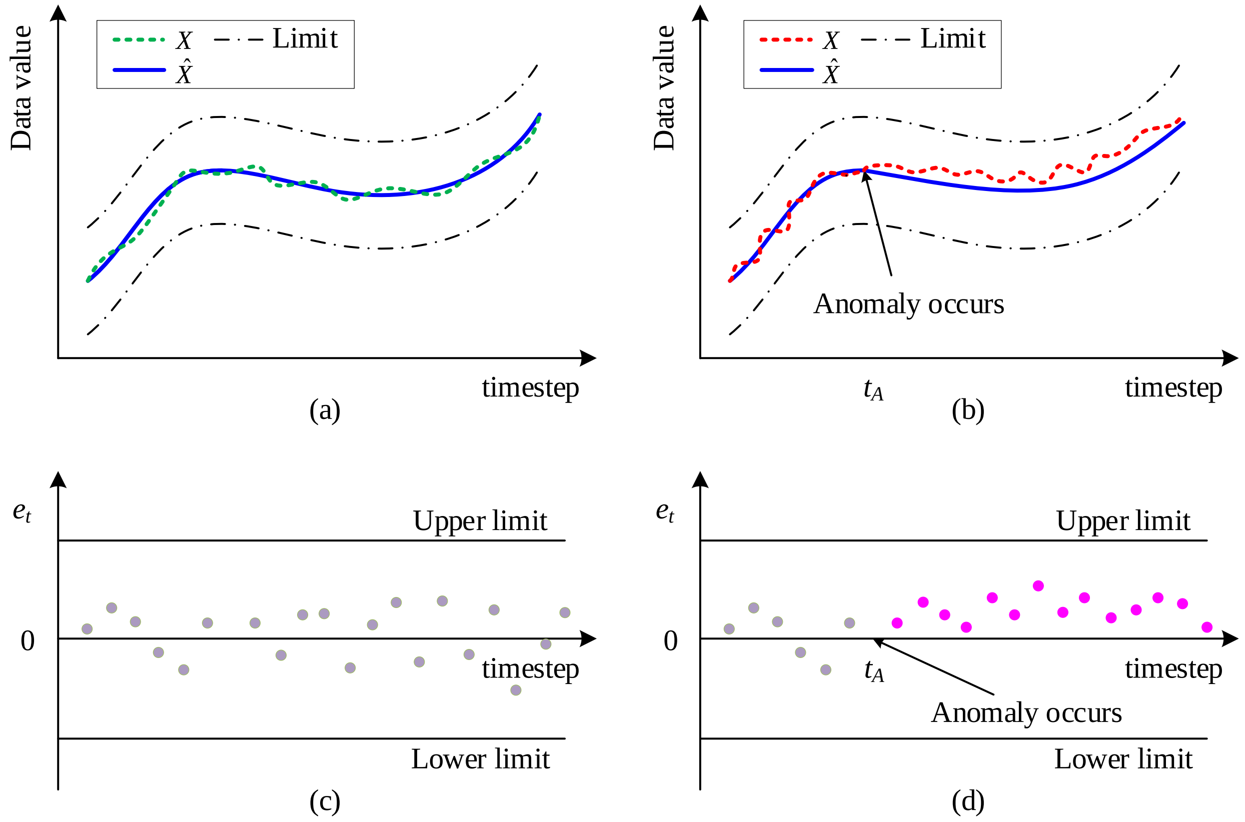


Figure 4.6. Architecture of the encoder-double decoder model.

4.2.4 Reconstruction Errors and Anomaly Detection Criteria

If there is no anomaly in the time series data, the reconstruction errors should vary randomly within certain limits. However, RNNs are generally sensitive to abrupt changes in the input values. Such an abrupt change of value may be routine, such as the engagement of a gearing system or a cutting tool encountering a workpiece, and not necessarily related to anomalous behavior or a quality problem. Only relying on extreme reconstruction error values as the criterion for an anomaly may lead to false positive (extreme value is associated with routine operation) or false negative (anomalous behavior does not manifest as an extreme value) signals. When an anomaly occurs, not only may the reconstruction errors increase, but the character of the reconstruction errors may also change, which may manifest itself in patterns/trends within the reconstruction errors. Thus, rather than only searching for singular extreme values as evidence of an anomaly, the reconstruction error series may also be monitored [61]. This is similar to using control charts to detect anomalies in a manufacturing process, in which not only are extreme points looked for in individual values, but also trends/patterns in the time series.

When anomalies occur, the magnitude of a reconstruction error may not be large, but it may show patterns/trends. For example, a time series, $\boldsymbol{X}$, is shown by the dashed green curve in Figure 4.7(a). The reconstructed time series, $\widehat{\boldsymbol{X}}$, is shown by the solid blue curve in Figure 4.7(a). The time series does not have an anomaly, and the reconstruction errors distribute randomly around the center line, as shown in Figure 4.7(c). The dashed red curve in Figure 4.7(b) corresponds to a time series, X, with an anomaly that occurred at time step t_A (the time series data are larger than values without an anomaly). Again, the reconstructed time series is shown as a solid blue curve. The magnitude of deviation is relatively small, and the reconstruction errors are within the limits, as shown in Figure 4.7(d). If only the magnitude of the reconstruction errors is considered, one might conclude that no anomaly has occurred. However, the reconstructed error data shows a pattern: after the anomaly, the errors are all positive. Such patterns in the reconstruction error data may also be used as criteria in anomaly detection.



(a) (b) (c) (d)

Figure 4.7. A pattern in reconstruction errors.

In a statistical process control chart, a series of tests are carried out to detect special causes/anomalies in time series data. For example, a set of successive points that run above or below the center line, a set of successive points that oscillate up and down, or a set of points showing continuous increases or decreases all constitute anomalous behavior (statistical signals) [87]. Knowledge and experience from a specific application may help to develop rigorous criteria for detecting trends in the reconstruction error data.

4.3 Experiments

Data from a diesel engine assembly process was used to assess the performance of the proposed anomaly detection model. The proposed model was trained using sensor data from routine operation of a diesel engine assembly line (anomaly-free data set). We show the benefits of using double decoders, and then show that the proposed model can detect anomalies in the sensor data from an assembly process subjected to three types of quality issues. The proposed models were implemented using Python and PyTorch.

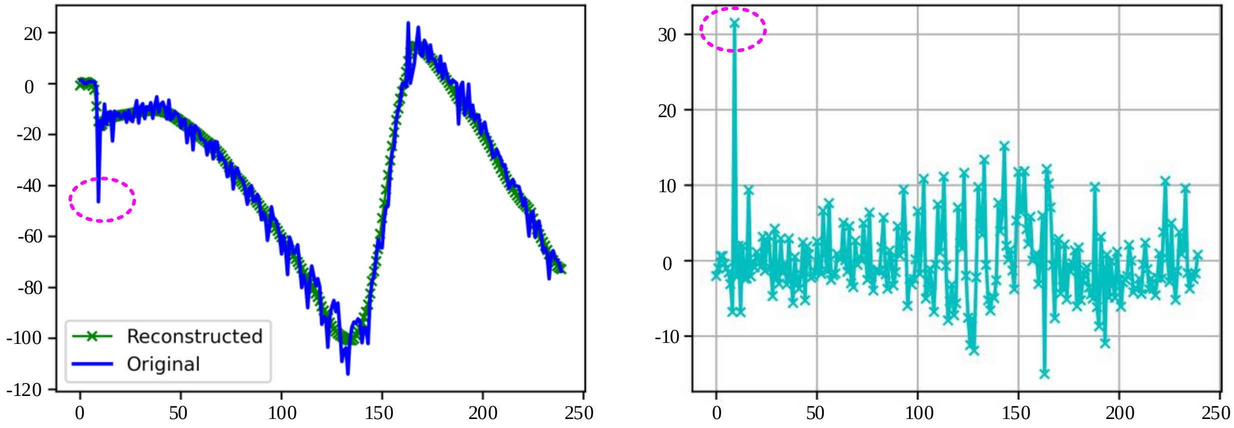
4.3.1 Time Series Data of an Engine Assembly and the Model Structure

The assembly of a diesel engine consists of a number of individual processes. One such process is the installation of the crankshaft into an engine block. Once the crankshaft is installed, several other steps must be undertaken, e.g., setting valve positions. During the valve position setting, the crankshaft is rotated. Time series data are collected during this rotation process. At each timestep, data containing measurements such as torque and rotational angle are collected to form a multidimensional vector. The data considers 240 timesteps, i.e., T=240, and the raw time series are the input to the encoder. To simplify the illustration, in Sections 4.3.2 to 4.3.3, we only considered anomalies related to torque, i.e., the input time series is one-dimensional. A multi-dimensional data set is used in Section 4.3.4.

As has been noted, abrupt changes in the time series data, may not necessarily be due to anomalies. For example, when an operator adjusts the height of a valve during crankshaft rotation, an abrupt change in torque value may result, but this change is not necessarily caused by a quality problem. Moreover, again as has been noted, anomalies may not manifest themselves as extreme values within the reconstruction errors.

In this experiment, LSTM was used in both the encoder RNN unit and the decoder RNN unit, with the hidden dimension being 512. The parameters are selected based on the literature and by experiments. Both the attention unit and the output unit were feedforward neural networks with two hidden layers. We built a training set using engines that have no assembly anomalies (noted as normal engines below). And then we trained the encoder-decoder attention model given in Chapter 4.2.2.1 using the training set.

Figure 4.8(a) shows the original time series and the time series reconstructed by the trained model. This was an engine without anomalies (quality issues) but with an abrupt change in the values (shown in the circle). The input time series data is not smooth, and the reconstructed time series follows the trend of the input time series, but it cannot fully reconstruct the changes of value with high frequencies. As shown in Figure 4.8(b), the reconstruction errors corresponding to the abrupt change is much higher (in the circle). This is because LSTM is generally sensitive to abrupt changes of value [61]. For the subsequence that is not smooth (the time steps between 100 to 150), the reconstruction errors are higher. But other than the peak that was caused by the abrupt change of the input time series data, no patterns are shown in the reconstruction errors.



(a) Original and reconstructed time series

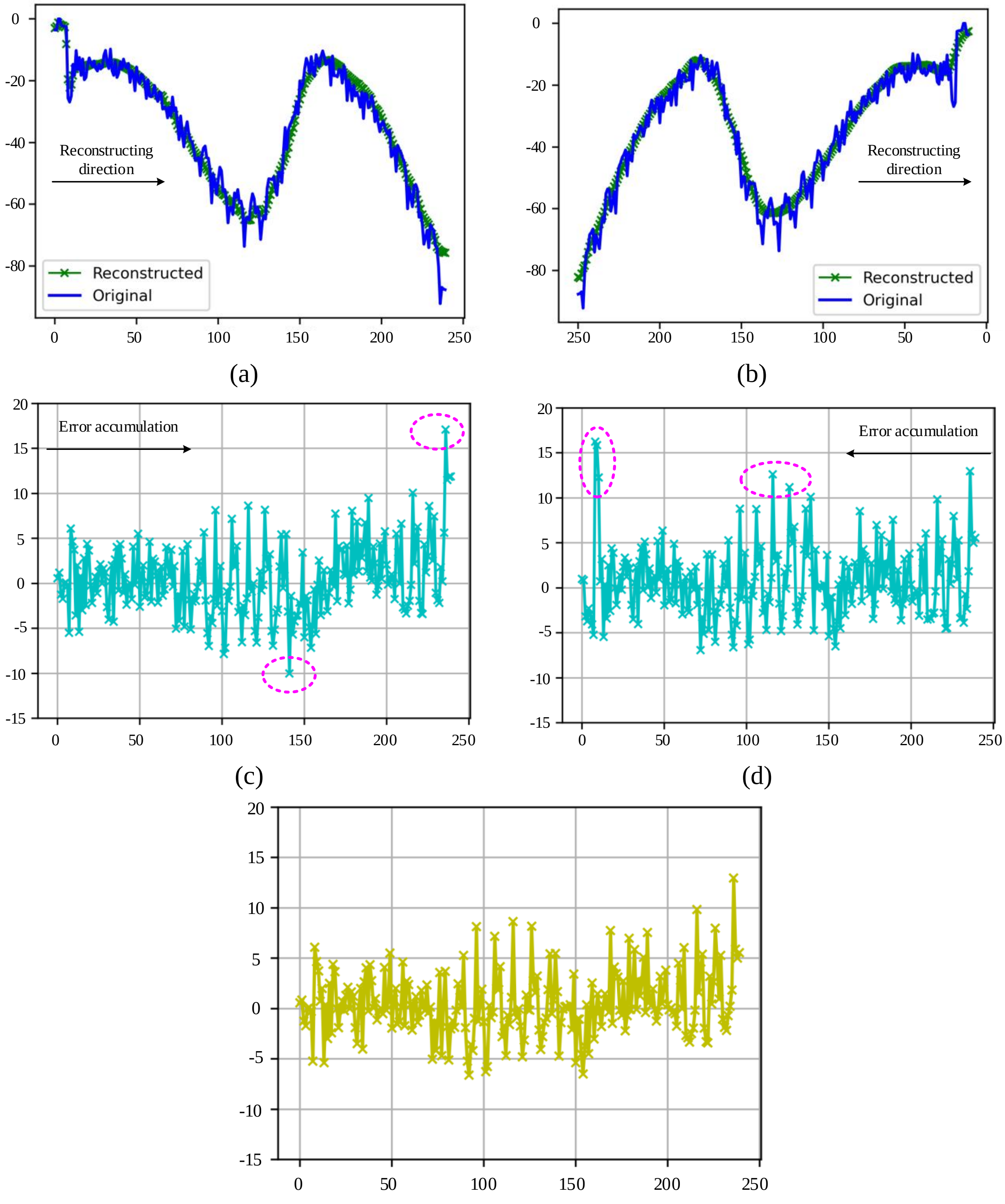
(b) Reconstruction errors

Figure 4.8. A time series data of an engine with no anomaly.

4.3.2 Double Decoders

In this section, we show the benefits of using two decoders. For the encoder-double decoder model, the parameters of the encoder RNN units, the attention units, and the output units are shared, respectively. The two decoders have the same structure but separate sets of parameters. We use the first decoder to reconstruct the data forward, from timestep 0 to 250, and use the second decoder to reconstruct the data backward, from timestep 250 to 0, as shown in Figure 4.9(a) and (b).

In Figure 4.9(c) and (d), we show both groups of reconstruction errors forward in timesteps to make it easier to compare them stepwise. The reconstruction errors accumulate with timesteps in the reconstruction direction (shown by the arrows in the figures). As highlighted by the circles in Figure 4.9 (c), when reconstructing data in the forward direction, there are a few reconstruction errors, seen later in the time series, that are much larger than the neighboring points. Similarly, as shown in Figure 4.9 (d), when reconstructing the data in the backward direction, there are a few reconstruction errors, seen early in the time series (later part of the reconstruction process), that are much larger than the neighboring points. For every timestep forward, the smaller ones of the two sets of reconstruction errors are included in a combined set of reconstruction errors, as shown in Figure 4.9 (e). As is evident from the figure, in the combined set of reconstruction errors, the extreme reconstruction errors are removed.



(a) (b)

(c) (d)

(e)

Figure 4.9. Reconstructed and original time series.

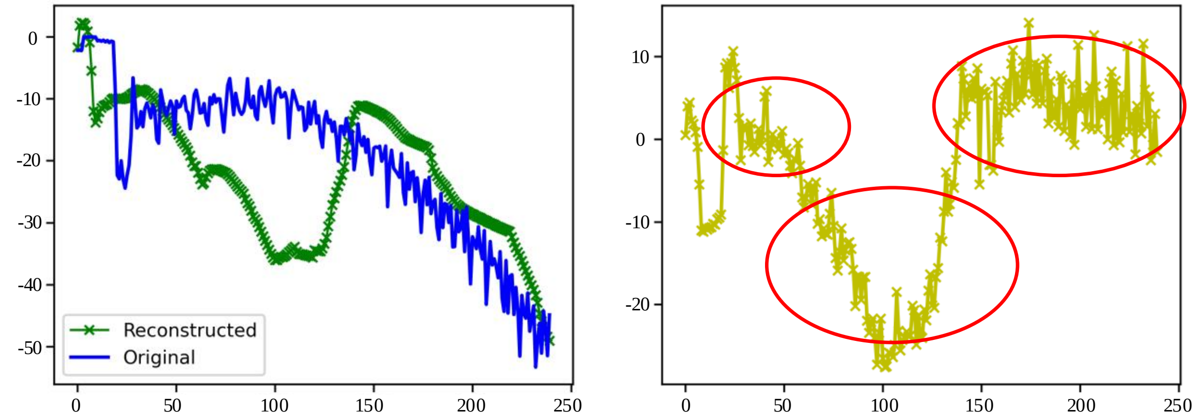
4.3.3 Anomaly Detection Results

Three types of common anomalies were considered, i.e., nonproper engagement of tools, stuck crank (e.g., nonsmooth change of torque caused by debris), and large torque. Four criteria were used to interpret the reconstruction errors: (1) A single point beyond the limits (33.50 and -43.12). (2) More than 12 points in a row on same side of the center line. (3) More than 11 points in a row, all increasing or all decreasing. (4) More than 14 points in a row, alternating up and down. Since there are few labeled anomalies, these criteria were first estimated by a validation set, and then tuned by analysis based on experience and criteria used in control charts [87]. Because there are few labelled anomalies, we tested one type of anomaly detected in the real manufacturing case and two types of anomalies that were manually introduced into anomaly-free time series data.

The first type of anomaly is caused by improper engagement between the gears of the driving tool and the crank of the engine. The shape of the time series with anomalies generally follows the shape of anomaly-free cases and with the maximum and minimum values being similar to anomaly-free time series. Therefore, it may be difficult for traditional anomaly detectors, which monitor general profile and extreme values of the time series, to detect the anomaly. The original and the reconstructed timeseries are shown in Figure 4.10 (a), and the reconstruction errors are shown in Figure 4.10 (b). The model successfully detected the anomaly. In the reconstruction errors, there are three subsequences shown through patterns (more than 12 points in a row on same side of center line), as shown in the three circles of Figure 4.10 (b).

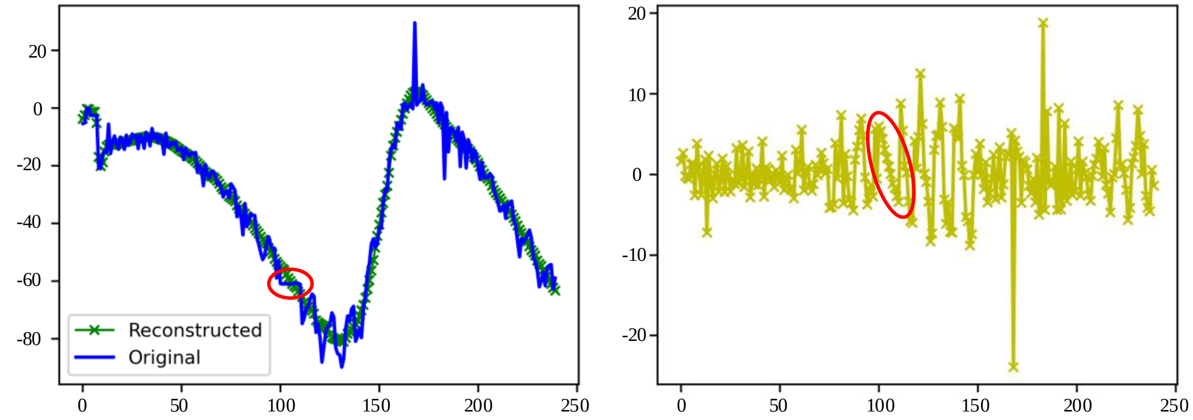
The second type of anomaly simulates the case of a crank that got stuck during rotation. The time series with anomalies is produced by modifying an anomaly-free time series. As shown in the circle in Figure 4.10 (c), the torques fluctuate near -60 for 10 timesteps. The model successfully detected the anomaly. As shown in the circle in Figure 4.10 (d), the reconstruction errors failed criteria 3 (more than eight points in a row, all decreasing).

The third type of anomaly simulates the case of missing components or extra components being assembled. With missing/extra components, the torque needed to rotate the crank may be different from the routine cases. For example, with an extra washer, it may need a higher torque, at each timestep, to rotate the crank. The time series with this anomaly is produced by scaling the torques of a normal engine by a factor of 1.3 (the maximum/minimum torque is still within limits). The input time series and the reconstructed time series are shown in Figure 4.10 (e). The model successfully detected the anomaly. The reconstruction errors failed criteria 2 (more than 12 points in a row on same side of the center line), as circled in Figure 4.10 (f).



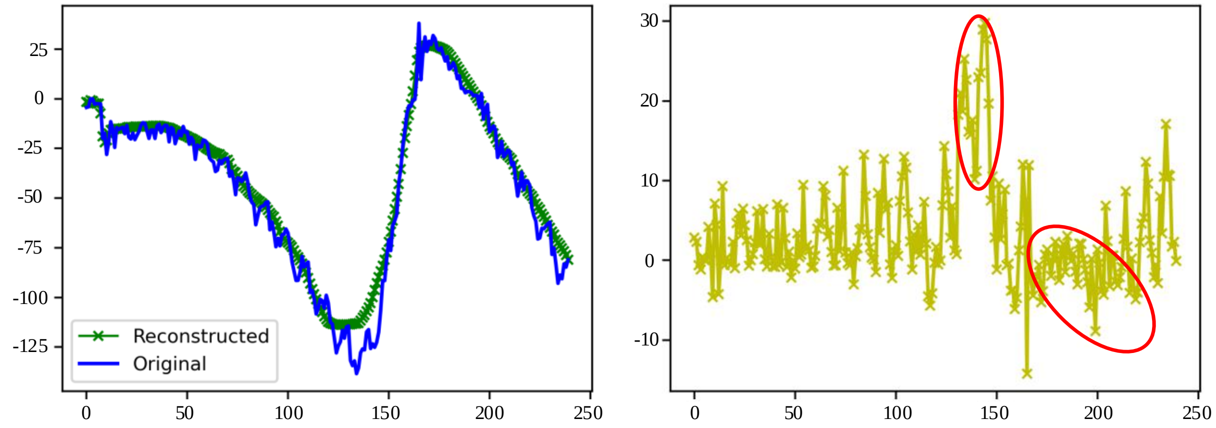
(a) (b)

Nonproper engagement of tools



(c) (d)

Stuck crank



(e) (f)

Large torque

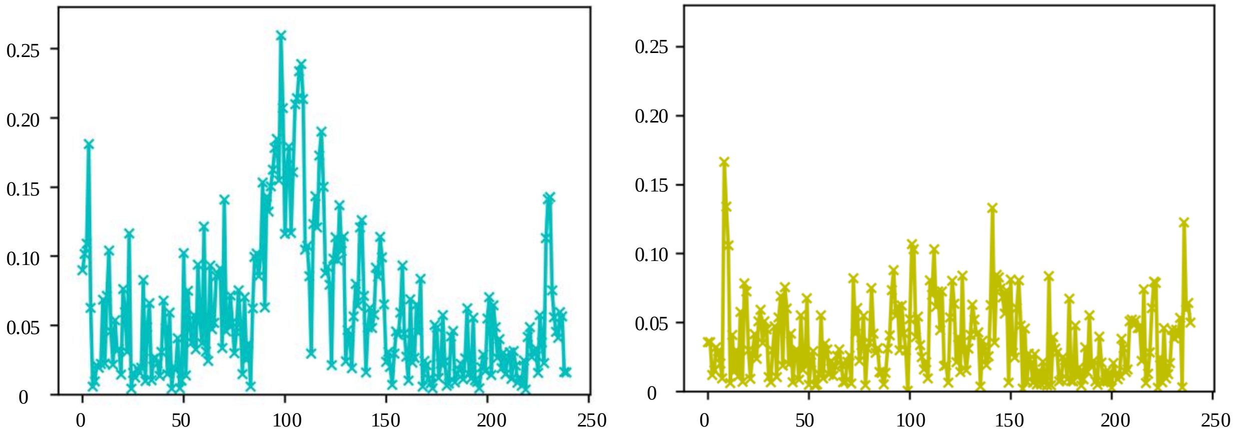
Figure 4.10. Three cases of detected anomalies. Left column: original and reconstructed time series. Right column: reconstruction errors.

4.3.4 Multi-Dimensional Time Series Data

This section compares the model proposed in this study to a model proposed by Malhotra et al. [56]. The comparison shows the advantage of the attention model as well as using two decoders. Experiments are also carried out to validate that the proposed model can detect anomalies in multi-dimensional time series data. As discussed in Section 4.1, for a multi-dimensional time series data, the Euclidean norm of the reconstruction errors, $\boldsymbol{e}$, can be used as the metric for detecting an anomaly. In such a case, the reconstruction errors at all time steps are positive. Experiments show that the anomaly detection criteria proposed in Section 4.2.4 are still applicable.

A two-dimensional time series data is considered. Other than the torque value used in the previous sub sections, time series data of rotational angles are also included. Results acquired by the two-dimensional data may be extended to data with higher dimensionalities. When the range of values in different dimensions varies too much, the data of a certain dimension may dominate the result. For example, in the two-dimensional time series data, the values of torque take on values from 0 to100, and the values of rotational angle vary from 0 to 2300. The Euclidean norm of the error vector will be dominated by the rotational angle. To avoid this situation, we rescaled the torque and rotational angles to be in the range of 0 to 1.

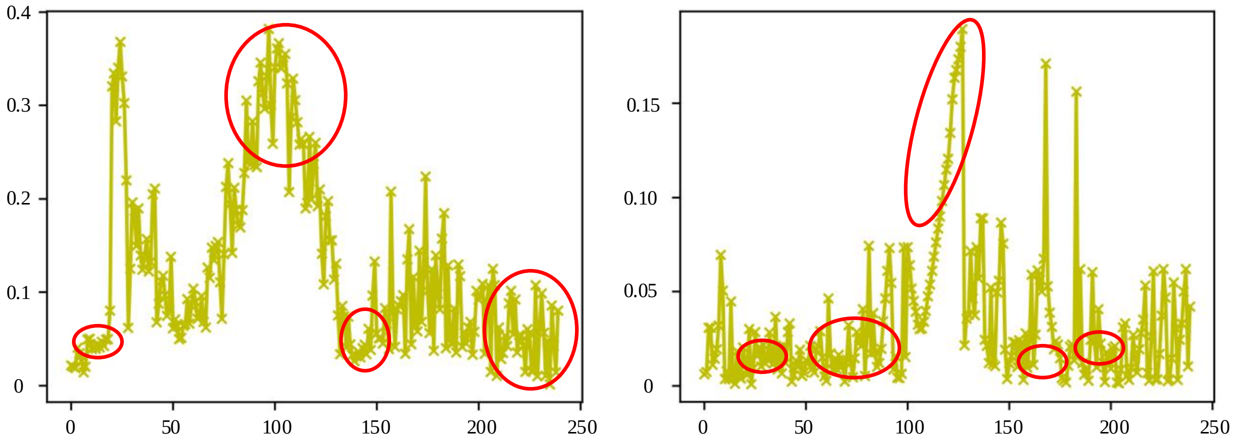
As mentioned in Section 4.2.2 and 4.2.3, compared to the traditional encoder-decoder model, the attention model and the structure of two decoders help summarize information in time series data and help attenuate the accumulation of reconstruction errors. Torque-rotational angle time series data with no anomalies are reconstructed by the traditional encoder-decoder model proposed by Malhotra et al. [56], and by the model proposed in this study (encoder-double decoder model with attention), as shown in Figure 4.11. It is apparent that the reconstruction errors for the model of this study are smaller. This means that the model of this study has a lower chance of making type I errors (false positives for anomaly detection).



(a) Malhotra et al. [56] (b) This study

Figure 4.11. Reconstruction errors of an anomaly-free time series.

Two examples are used to validate the effectiveness of the proposed model in detecting anomalies in multi-dimensional time series data. There are anomalies in both examples, and the model successfully detects the presence of these irregularities. The limits for magnitude and trends in the multi-dimensional data are the same as given in Section 4.3.3. The reconstruction errors of the first example are shown in Figure 4.12 (a). The points in the circles are successfully detected anomalies where there are more than 11 points in a row, all increasing or all decreasing. The reconstruction errors of the second example are shown in Figure 4.12 (b). The points in the circles are anomalies where either there are more than 11 points in a row, all increasing or all decreasing, or there are 11 points in a row on same side of center line.



(a) Time series one

(b) Time series two

Figure 4.12. Reconstruction errors of two time series data with anomalies.

4.4 Conclusions

New technologies such as smart sensors and internet of things have enabled manufacturers to accumulate a large amount of data collected from manufacturing systems, which may take the form of a multidimensional time series of long length. This chapter has proposed a new model that can detect anomalies in a time series data, which usually contain useful information of the manufacturing system such as failures and defects. Detecting these anomalies enables manufacturers to catch failures and defects early, and to take remedial actions and precautions.

The proposed model is based on the encoder-decoder reconstruction model. The model takes time series data as an input and reconstructs the input time series. The differences between the input time series data and the output time series data are analyzed to detect anomalies. Both the encoder and the decoder are recurrent neural networks. An attention model was incorporated into the reconstruction model, which improved the model's ability to summarize information from the time series data. Based on the attention model, an encoder-double decoder model was proposed. Using two decoders decreases the accumulation of the reconstruction errors. A few anomaly detection criteria that were inspired by statistical process control techniques were introduced to detect trends in the reconstruction errors. These criteria can be used in conjunction with the limits for the reconstruction errors in detecting anomalies. The performance of the proposed method was assessed using data from a diesel engine assembly process. Three common types of anomalies were successfully detected from the time series data. It was also shown that the method can not only detect anomalies, but it may also provide information on the timestep at which the anomaly occurred. This feature allows a manufacturer to pinpoint the source of the problem.

The major contributions of the chapter are

- An attention model, which was used in machine translation, was adapted, and incorporated into the encoder-decoder model. The attention model enables the encoder to better summarize the content in the input time series by automatically detecting which parts of the input data are more relevant in reconstructing the data at each time step.
- A double decoder model was proposed. Using two decoders attenuates the accumulation of reconstruction errors. The two decoders share model units such as the encoder, which greatly reduced the parameters in the training process.
- The introduction of additional criteria in interpreting the reconstruction errors. Other than only using extreme values of reconstruction errors as evidence of an anomaly, we introduced the idea of using trends in the reconstruction errors as criteria of an anomaly. Such a technique decreases the false alarms and improves the capability of the model to detect multiple types of anomalies.

The fact that the proposed model can be trained using data acquired during routine system operation makes it very practical. Often, there are few data labeled as anomalies, since anomalies are generally rare cases in a manufacturing system that is under control. However, the current model needs human experience in tuning the criteria used in monitoring the trends of reconstruction errors.

Future research such as statistical analysis may be carried out to automatically set the criteria used to interpret the reconstruction errors. The structure of the proposed neural network and some of the hyper parameters such as learning rates are selected based on experience and experiments, studies may be carried out to optimize these parameters. Two-dimensional data are used to validate the applicability of the model for multi-dimensional data. Like control charts can be applied to many manufacturing processes and systems, the proposed methods can likewise be applied in many circumstances. Readers are encouraged to test the model using data with even higher dimensions.

5. COMPONENT-ORIENTED REASSEMBLY IN REMANUFACTURING SYSTEMS: MANAGING UNCERTAINTY AND SATISFYING CUSTOMER NEEDS[5]

Remanufacturing has recently received significant interest due to its environmental and economic benefits. Traditionally, the reassembly processes in remanufacturing systems are managed using a product-oriented model. When a product is returned and disassembled, the used components may be processed incorrectly, and the quality of the remanufactured products may not meet customer needs. To solve these problems, a component-oriented reassembly model is proposed. In this model, returned components are inspected and assigned scores according to their quality/function, and categorized in a reassembly inventory. Based on the reassembly inventory, components are paired under the control of a reassembly strategy, and these pairs are then assembled into reassembly chains. Each chain represents a product. To evaluate the performance of different reassembly strategies under uncertain conditions, we describe the reassembly problem using an agent-environment system. The platform is modeled as a Markov decision process, and a reassembly-score iteration algorithm (RSIA), is developed to identify the optimal reassembly strategy. The effectiveness of the method is demonstrated via a case study using the reassembly process of diesel engines. The results of the case study show that the component-oriented reassembly model can improve the performance of the reassembly system by 40%. Sensitivity analysis is carried out to evaluate the relationship between the parameters and the performance of the reassembly system. The component-oriented model can reassemble products to meet a larger variety of customer needs, while simultaneously producing better remanufactured products.

---

[5] Reprinted with permission from (portions enhanced/adapted) Y. Wang, G. Mendis, S. Peng, and J. Sutherland, "Component-Oriented Reassembly in Remanufacturing Systems: Managing Uncertainty and Satisfying Customer Needs," J. Manuf. Sci. Eng., vol. 141, no. 2, pp. 0210051–02100512, 2019.

With increased awareness of the environmental issues attributable to manufacturing processes and products, researchers have started to investigate methods to make manufacturing processes and products more sustainable. Remanufacturing is one of these efforts that closes the loop in a product life cycle. As defined by Ortegon et al. [128], remanufacturing is a process that preserves the functional values of end-of-life products that are added in the design and manufacturing stages of the product life cycle. By transforming end-of-life products back to use-phase products, the material, energy, and labor that was originally invested in the products can be reused, thus decreasing the environmental burden of the product life cycle. In addition to these environmental benefits, remanufacturing has many other benefits. For example, remanufacturing is usually a faster and more cost effective way to customize products, to maintain products by replacing bad components, and to develop products that satisfy a variety of customers' needs [129].

Products, such as engines, are complete systems that are sold to customers. Products are made up of components, which enable the product to perform its purpose. While the distinction between a component and a product is somewhat arbitrary, a product is always the more complex system. A product has a set of functions; for instance, an engine provides mechanical energy to a rotating output shaft. Products/components also have functional levels, which are the degrees to which a product/component performs specified functions. In the engine example, an engine might be designed to produce 600 kW of maximum power, but after years of usage, may only be able to produce 500 kW of maximum power. The quality level of a component refers to specific characteristics that often relate to function. For example, the quality level of an engine block could be related to a characteristic such as the average diameter of the cylinders. The functional level of a product is dependent on both the functional level and the quality level of the components in the product. Failure to recognize this fact can lead to lost and wasted value during remanufacturing.

This chapter proposes a component-oriented model to control the reassembly process of a remanufacturing system by taking the uncertainties of returned components and the variety of customer needs into consideration. In this model, returned components are inspected, a score is assigned to each component according to its quality and/or functional level, and the components are organized in a reassembly inventory. Reassembly strategies are developed using an agent-environment system. The reassembly process is modelled as a Markov decision process (MDP). A reassembly-score iteration algorithm (RSIA) is developed to identify the optimal reassembly strategy and a Monte Carlo method is used to deal with uncertainties. These methods are applied to a case study of the remanufacturing of a diesel engine, and sensitivity analysis is performed to examine the performance of the model in comparison with an exhaustive reassembly strategy.

5.1 Component-Oriented Reassembly

A remanufactured product consists of components, and for each component, a quality or function score may be estimated based on its quality level or functional capability. The specifics of the quality and function assessment are outside the scope of this work, but a number of researchers have investigated methods for inspecting and evaluating products [78] [79] [130]. While quality and functional level may express two different attributes of a component, here we assume that the inspection process can assign a single score to a component, based on its quality level or functional capability. By choosing returned components with different functional capabilities (scores), different requirements can be quantitatively achieved using a strategy that governs the reassembly process of the system (hereafter called the reassembly strategy).

5.1.1 Reassembly System

The reassembly system consists of four major processes, as shown in Figure 5.1. First, returned products are collected. Then the products are dismantled into components. An inspection process gives a score to each returned component according to factors such as appearance, service years, and presence of cracks, corrosion, pitting, and wear. Components with scores below a given threshold will be recycled, while components that are worth remanufacturing will be sorted into the reassembly inventory according to their scores. Some of these components may be reconditioned/remanufactured (the component is then rescored), and some new components may be added to the inventory. Finally, the reassembly process is carried out under the control of a reassembly strategy. Each reassembled product is then assigned a product score, which is based on the scores of all the components in the product. The performance of the reassembly system can be evaluated using the total scores of all the reassembled products.

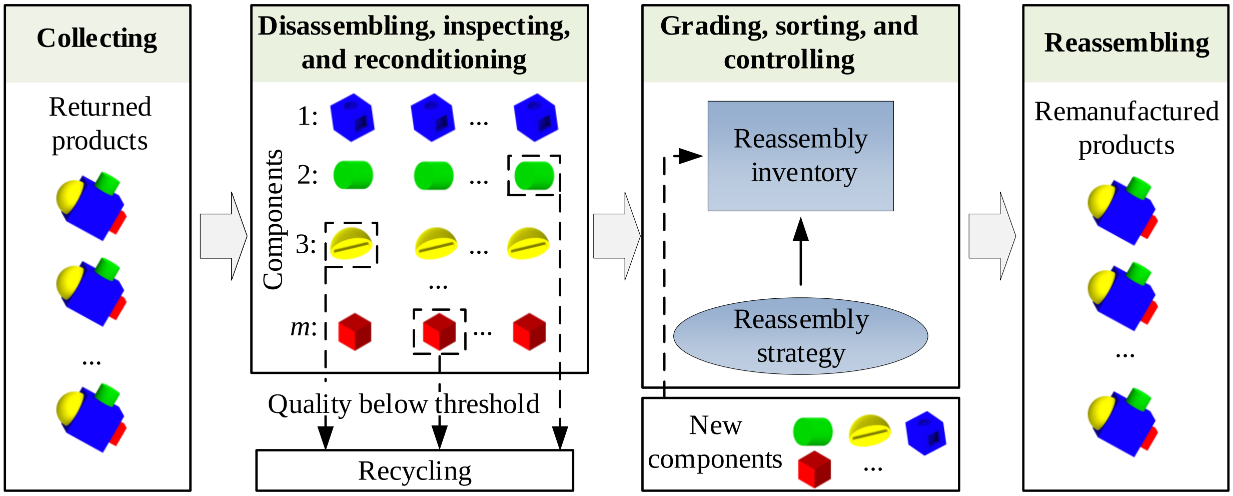


Figure 5.1. Reassembly system.

5.1.2 Reassembly Processes

When returned products are disassembled, inspection is necessary to determine the scores of the components. These scores will be used in subsequent steps.

5.1.2.1 Inspection

One of the challenges in developing a remanufacturing system is characterizing the quality and/or functional level of returned components. Let us assume that a component's actual functional performance is described by a true score (TS). In practice, however, this TS is characterized through an inspection operation that provides a measured score (MS). The obtained measurement score is subject to measurement uncertainty and is therefore a random, continuous variable [131]. While precise measurement could be used to best estimate the actual TS, this can be difficult and costly. Often, simple, less precise, and less costly measurement technologies may be used. And, as has been noted, once a component has been measured, it may be placed into grades or categories (C_1, C_2, C_3, etc.), with each category having an associated nominal value of c_1, c_2, c_3, etc. (often, interval mid-points). The category score (CS) for a given MS is the value of c_i corresponding to the appropriate category. The value of CS is discrete. The three types of scores are described in Table 5.1.

Table 5.1. Summarization of Scores.

Score	Purpose	Value
True score (TS)	True quality level of a returned component	Single
Measured score (MS)	A measurement that seeks to characterize component quality. An estimate of TS.	Continuous
Category score (CS)	Categorization of MS	Discrete

Once a component is placed in a category, there is no memory of its original measurement, MS. CS is an estimate of the TS for a component. The relationships among TS, MS, and CS are illustrated by an example in Figure 5.2. In this case, a specific component is drawn from the measurement distribution and has a value equal to r. This specific measured component is then assigned to category C_2, which has a midpoint of c_2. Thus, CS=c_2 for that specific component. It is assumed that the measurement distribution is unbiased (i.e., E[MS]=TS). Due to measurement uncertainties, there are possibilities that the CS value for this component could have been c_1 or c_3. The probability of such occurrences may be calculated using the measurement distribution. This uncertainty is considered in the reassembly strategy in Chapter 5.2 and is estimated by simulation in the case study.

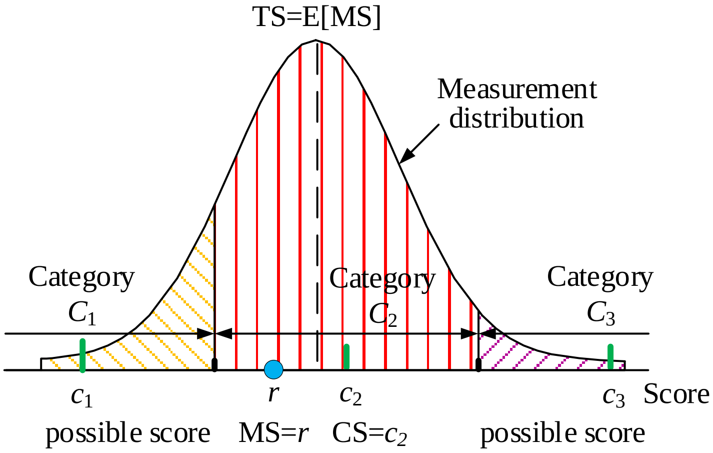


Figure 5.2. An example of the relationship among TS, MS, and CS. (unbiased measurement distribution).

5.1.2.2 Reassembly Inventory and Reassembly Strategy

A reassembly inventory is used to manage the reassembly process. The reassembly inventory stores the components based on their corresponding category scores (CS values). A reassembly inventory can be modelled as a table with m rows and n columns, in which m is the number of components required to reassemble a product and n is the number of different categories. n can be determined by remanufacturers using a variety of methods, for instance, by surveying customers [65] [132], or by using historical sales data of remanufactured products. Each component will be categorized into a cell of the reassembly inventory according to its component type and score. For ease of illustration, we assume that each cell of the inventory has at most one component.

In the reassembly inventory, when a component with a certain CS does not exist, a new component can be purchased to fill the gap. The quality score of a new component is greater than or equal to the highest quality score of a returned component in the reassembly inventory. This choice on whether to buy a new component is controlled by the reassembly strategy, which takes total costs/benefits of the reassembly system into consideration.

Each product can be viewed as a reassembly chain, in which *m* components are grouped. The scores of components in a reassembly chain determines the score of this chain. The summation of scores of all reassembly chains, noted as the total score of the inventory, can be used to evaluate a given reassembly strategy.

Figure 5.3 shows a reassembly inventory and several possible reassembly strategies corresponding to different scores. The task of identifying the optimal strategy to reassemble components is transformed into the task of finding the optimum strategy of building reassembly chains. It should be noted that the order of the components in the reassembly chain does not necessarily correspond to the physical connections/relationships in the reassembly processes.

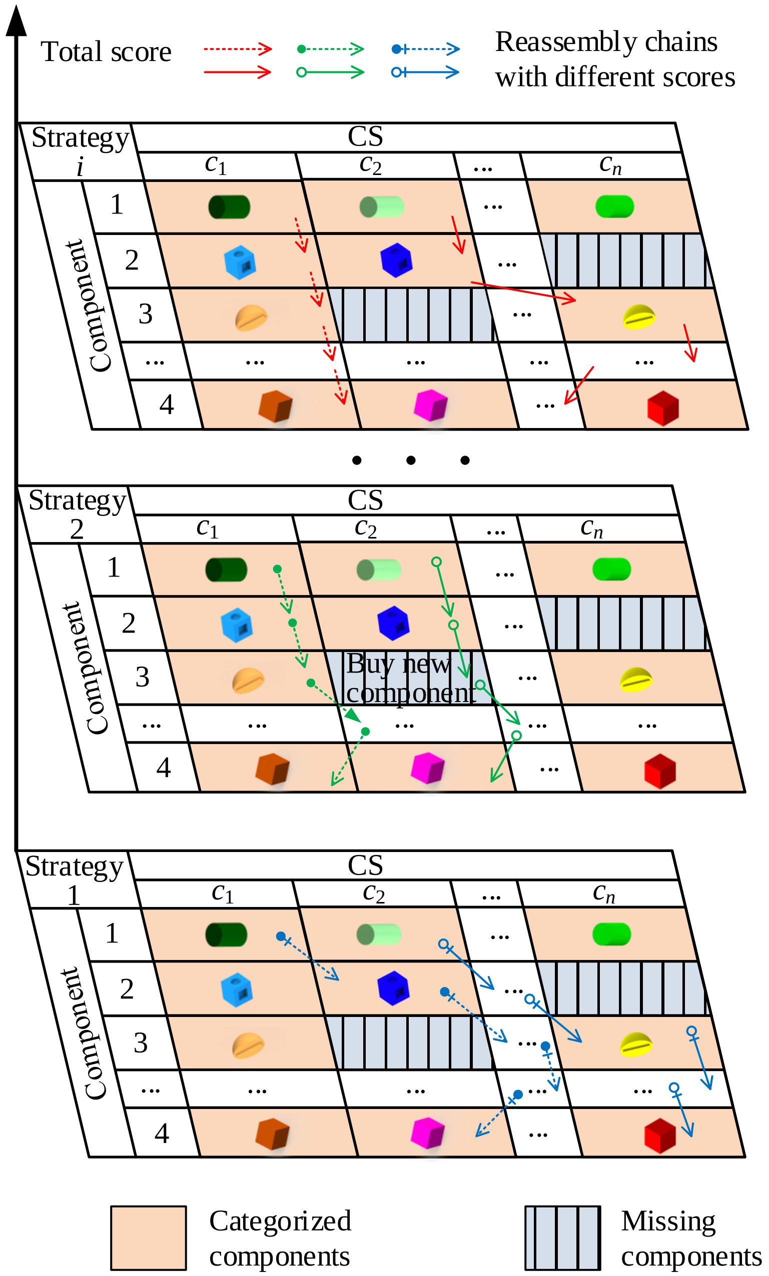


Figure 5.3. A reassembly inventory and multiple reassembly chains.

5.1.2.3 Building a Reassembly Chain

Working with the reassembly inventory becomes complicated when multiple types of components are scored. And the uncertainties of the quality/functional levels of components make it very hard to find the reassembly chains corresponding to the highest total score. To tackle this challenge, a reassembly chain is viewed as links of component pairs: by grouping components from the reassembly inventory into component pairs, and by linking the component pairs into a reassembly chain, a product is reassembled. The structure of reassembly chains is adjusted by pairing, which is a series of processes of selectively grouping two different types of components that will be reassembled into the same product.

Figure 5.4 shows the CSs and corresponding measurement distributions for two types of hypothetical components. The upper plot and the lower plot correspond to the green cylinder and red cube components, respectively, from the example product in Figure 5.3. In the figure, different shades of color refer to components placed in different categories (the measurement distributions are different). In this illustrative example, four groups of components are paired under the control of a certain reassembly strategy. The components will be assembled into four products (together with other components, which are not shown in this figure).

If a product is made from *m* components, then (*m*-1) steps of pairing processes are needed, and each step of the pairing process affects the following steps, because the score of the products is determined by all the links in the chain.

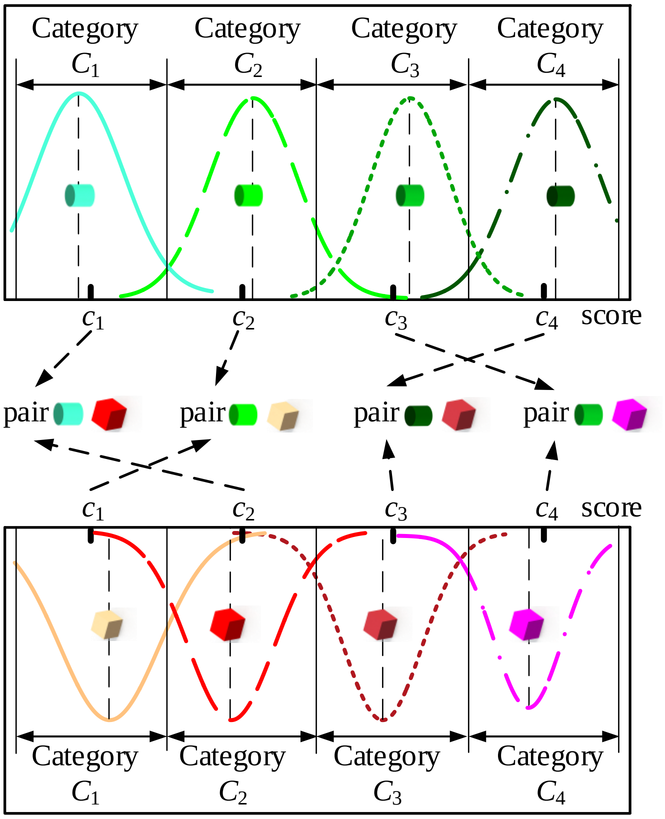


Figure 5.4. An example of pairing processes.

5.1.2.4 The Objective Function

An objective function is used to evaluate a reassembly chain according to the requirements of the manufacturer. The score of the *k*th product, W_k, is computed using a function *F*. A general form of *F* is shown below:

$$W_k = F(\{c_{i,j}\}), i = 1, 2...m, j \in \{1, 2...n\}, \tag{5.1}$$

where $\{c_{i,j}\}$ represents the set of all components' CSs in a reassembly chain. The total score of the reassembly inventory is calculated by the following equation:

$$W = \sum_{k=1}^{u} W_k \text{ ,} \tag{5.2}$$

which is the summation of the scores of all the reassembled products. u is the number of products that are reassembled, and the value of u is less than or equal to the number of columns of the reassembly inventory, i.e., $u \leq n$.

Different objectives, such as minimizing the number of wasted components, minimizing the cost of the system, and increasing the quality of reassembled products, can be implemented by adjusting the objective function in Equation (5.1). Two specific forms of the objective function are given below, and other forms may be designed according to needs.

FORM ONE: COMPONENT PAIR-BASED FUNCTION. If the physical condition of a component, such as its dimensional accuracy or wear is used as a major indicator of the score, mating relationships between the components (for example, two components that are in contact with each other) can be considered in the objective function. Some studies provide methods to grade used components based on dimensional variations [77] [133]. The score of the *k*th product is defined as the summation of the scores of all the component pairs, and the score of a component pair is the lower score of the two mating components:

$$J_i = F(\{c_{i,j}\}) = \mathrm{Min}(\{c_{i,j}, c_{i+1,k}\}), i = 1, 2 \ldots m-1, j \in \{1, 2 \ldots n\}, k \in \{1, 2 \ldots n\} \text{ .} \tag{5.3}$$

$$W_k = F(\{c_{i,j}\}) = \sum_{i=1}^{m-1} J_i \text{ .} \tag{5.4}$$

In Equation (5.3), J_i is the score of the component pair, $\{c_{i,j}, c_{i+1,k}\}$.

FORM TWO: WEAKEST COMPONENT-ORIENTED FUNCTION. The remaining useful life (RUL) of a remanufactured product is one of the most important metrics to estimate the value of the product [69]. The RUL of a product heavily relies on the RULs of its components. Some methods have estimated the RUL of the returned components [134][135][136], so the RULs can be used as the CSs of the components. The score of a reassembly chain is an estimation of the RUL of the reassembled product. We can design the objective function F using the weakest link principle: the RUL of the product is determined by the component with the lowest RUL in the reassembly chain. The score of the kth product can be computed using Equation (5.5):

$$W_k = F(\{c_{i,j}\}) = \mathrm{Min}(\{c_{i,j}\}), i = 1, 2 \ldots m, j \in \{1, 2 \ldots n\}. \tag{5.5}$$

These two forms of the objective function can be used to optimize reassembled products for different business scenarios. To achieve the best business results, the reassembly strategy, which controls the reassembly process, should be optimized.

5.2 Optimal Reassembly Strategy

For a given objective function, reassembly chains with optimal combinations of components should be built. For any given reassembly inventory, there exists an optimal reassembly strategy with a corresponding total score, W, that is not lower than any other reassembly strategy. To find the optimal reassembly strategy considering the uncertainty in the scores of components, we model the reassembly process using the agent-environment system. The agent-environment system is a powerful tool to evaluate different strategies by estimating the current state of the system and potential improvements [137]. The platform is modeled as a Markov decision process. A reassembly-score iteration algorithm (RSIA), which is an algorithm based on dynamic programming, is used to analyze the agent-environment system, and find the optimal reassembly strategy.

5.2.1 Agent-Environment System and Optimal Reassembly Strategy

In the agent-environment system, the agent and the environment interact in a sequence of "time steps." At each time step, the agent selects from the possible actions under the control of a policy. At time step t, the state of the environment is represented as S_t. For a finite state problem, the set of all possible states is noted as $\boldsymbol{S}$, and $S_t \in \boldsymbol{S}$. At each state, the agent takes an action A_t, the set of all available actions is noted as $\boldsymbol{A}$, and $A_t \in \boldsymbol{A}$. After each action, the agent receives a reward $R_{t+1} \in \boldsymbol{R}$ and will be in a new state S_{t+1}. The agent evaluates the state, S_t, it is in, and uses policy, π_t, to determine what action to perform. In the agent-environment system, the agent achieves the goal of maximizing the rewards it will receive over the long run.

The reassembly system can be viewed as an agent-environment system, as shown in Figure 5.5. The reassembly decision maker is the agent, and the reassembly inventory is the environment that the agent interacts with. The current pairing relationships between the components in the reassembly inventory is the state S_t. Changing the current pairing relationships from state S_t to state S_{t+1} is action A_t. The reassembly strategy is the policy $\pi_t(A=a|S=s)$, which controls the reassembly process according to the current pairing relationships. The reward R_{t+1} is the change of product score due to the change of the pairing relationships. It should be noted that the time step t is used to distinguish states and actions; it does not necessarily correspond to physical time. The agent-environment system enables iterative evaluations of the reassembly system.

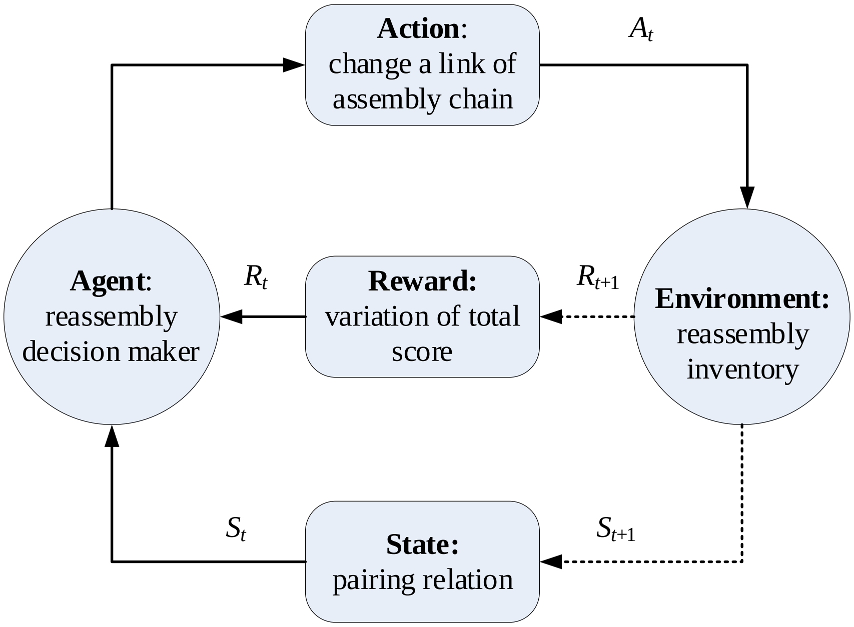


Figure 5.5. Agent-environment system for a reassembly system, adapted from Sutton and Barto [29].

The goal of the reassembly decision maker, i.e., the agent, is to accumulate the most rewards from the reassembly processes over the long term (as opposed to the short term). This is achieved by finding the optimal reassembly strategies to pair the components in the reassembly inventory so that the total score of the products assembled from the reassembly system is maximized. Maximizing the reward is equivalent to achieving the objectives that are embedded in objective function F of Equation (5.1). Using this notation, the process is performed by evaluating the expected future rewards of being in state S_t, then carrying out action A_t, as controlled by policy π_t, until an optimal reassembly strategy is found.

The expected accumulated rewards G_t can be computed using the following equation:

$$G_t = (R_{t+1} - D_{t+1}) + \gamma(R_{t+2} - D_{t+2}) + \gamma^2(R_{t+3} - D_{t+3}) + \ldots = \sum_{k=0}^{\infty} \gamma^k (R_{t+k+1} - D_{t+k+1}), \qquad (5.6)$$

in which D_t is a number that represents all the negative effects of the current pairing relationships. Remanufacturers can assign values to D_t, in order to consider costs in their remanufacturing systems, for example, inventory costs, inspection costs, maintenance costs, handling costs, and so on [138]. γ is the discount rate, and $0 \leq \gamma \leq 1$. The discount rate determines whether the agent is shortsighted or farsighted: when $\gamma = 0$, only the immediate (shortsighted) reward is considered; when $0 < \gamma < 1$, the future reward is considered less valuable than the current reward; when $\gamma = 1$, the future reward is equally valuable as the current reward. In this reassembly agent-environment system, because the time t does not correspond to physical time, $\gamma = 1$ is used, so that the maximum total score can be found.

As the goal of the reassembly decision maker is to accumulate the highest rewards from the reassembly processes in the long run, the long-term reward, G_t, equals the expected total score of the reassembly inventory, $W(S_t, A_t)$:

$$W^*(S_t, A_t) = E[R_{t+1} - D_{t+1} + \max\{W^*(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\}]. \tag{5.7}$$

For a given timestep, the total score of the reassembly inventory under the control of optimal reassembly strategy, π^*, can be evaluated by the optimal reassembly value function, $W^*(S_t, A_t)$, which is the highest score that the given reassembly system can achieve. $W^*_n(s, a)$ can be computed by the following Bellman optimality equation:

$$W(S_t, A_t) = E[G_t \mid S_t = s, A_t = a]. \tag{5.8}$$

Equation (5.8) is a system of T nonlinear equations, in which each state of the reassembly inventory corresponds to an equation, and the number of variables can be computed using the following equation:

$$T = (n!)^m, \tag{5.9}$$

in which, m is the number of components, and n is the number of CS categories. Equation (5.9) does not consider cases when missing components exist; in this scenario, there will be more states to consider.

The optimal reassembly strategies are found when Equation (5.8) is solved. However, it is very hard to find closed form solutions of the Bellman optimality equation for the reassembly system, for three major reasons:

1) The expected value of the total score is very difficult to calculate because of the uncertainties inherent in the inspection processes.
2) The scale of the equation system may be too large to solve if the total number of possible states T increase. For example, if a reassembly inventory with five components and six scores, there will be more than 10^{12} unknown variables.
3) The equations are nonlinear.

Due to the reasons mentioned above, instead of using an analytical method, we develop a reassembly-score iteration algorithm (RSIA) to solve the optimal reassembly-value function, $W^*(S_t, A_t)$.

5.2.2 Reassembly Score Iteration Algorithm

In this section, a RSIA is proposed to find the optimal reassembly strategy. It is based on a value iteration algorithm, which is an iterative method used to find the optimal policy of MDP models [137] [139]. As can be proved using Bellman optimality equation, the RSIA converges to the optimal value function $W^*(S_t, A_t)$. In the iteration process, the reassembly-value function is updated using Equation (5.10):

$$W_{k+1}(S_t, A_t) = \max_{a \in \boldsymbol{A}} \left(E[R_{t+1} - D_{t+1} + W_k(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] \right), \tag{5.10}$$

in which k is the iteration step. One way to compute the expected value in Equation (5.10) is to multiply the possible score by various probabilities of obtaining such a score. However, at issue is estimating these probabilities.

As has been noted, every component is measured (distribution of MS is assumed normal with variance σ^2), and the measured value, r, allows the component to be placed in a category, C_i, where it assumes the value CS=c_i. Once a component is categorized, no memory remains of the measured value, r. For a given CS, and depending on the measurement uncertainty, there is a probability that the component has any TS. Perhaps CS is very close to TS, or it could be that CS is very different from TS. While this latter case is unlikely, it could occur (especially when measurement uncertainty is large) because the probability of having a TS very different from c_i, is greater than zero. To estimate these probabilities, an understanding is needed of: i) how TS may differ from component to component, and ii) the measurement distribution. With such information, the probabilities of TS aligning with a category or a different category for a given CS may be determined analytically or estimated from a Monte Carlo simulation. Essentially, for each component, a table is needed that shows the probability of being in each range of TS (TSs are categorized to integers to simplify computation) given the CS of the component. An illustrative example shows this situation in Figure 5.6. In this example, components are placed into three categories.

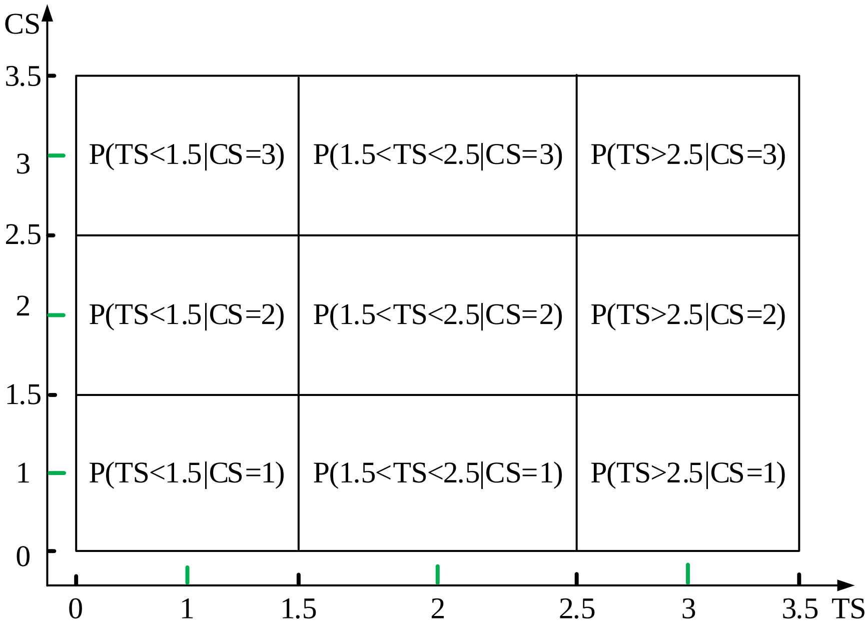


Figure 5.6. Probability table for a three-level categorization.

Based on the probability table, the expected value of score, $W_{k+1}(S_t, A_t)$, can be computed. Since each component in a reassembly chain has a measurement uncertainty, the average score is used: action A_t is carried out H times at each iteration, and the average total score for these H actions is the expected total score. When the value of H is large enough, the average total score converges to the expected value of the total score for the given reassembly strategy.

As presented in Section 5.1.2.2, the task of finding the optimal strategy to control the reassembly system is equivalent to finding the optimum strategy of grouping components from the reassembly inventory into reassembly chains. The RSIA is built based on a data structure that uses the pairing relationships of the reassembly chains. This approach simplifies the search process by adjusting the pairing relationship between the sub chains.

It is to be noted that $T^z(i,j)$ is the component in the ith row and jth column of the reassembly inventory, and the component is grouped in the zth reassembly chain. For component $T^z(i,j)$, the value of $T^z(i\text{-}1,k)$ and $T^z(i+1,h)$ are stored, which are the positions of the components in the row above and the row below it, respectively, in the reassembly inventory. By doing this, a reassembly chain is stored as a group of links and can be dismantled at any link without changing the connections elsewhere in the chain.

The details of the proposed RSIA are shown in Table 5.2.

Table 5.2. Reassembly score iteration algorithm.

This algorithm builds the optimal chains of the reassembly inventory

---

Input:

m:	The number of rows of the reassembly inventory
n:	The number of columns of the reassembly inventory
$\boldsymbol{U}$	Chaining information. $\boldsymbol{U}=\{U_{x,y}\}$, $x=2,3,\ldots,m$. $y=1,2,\ldots,n$. $U_{x,y}$ is the column index of the component in row $(x-1)$, to which the component with coordinate (x,y) pairs.
$\boldsymbol{B}$	Chaining information. $\boldsymbol{B}=\{B_{v,z}\}$, $v=1,2,\ldots,m-1$. $z=1,2,\ldots,n$. $B_{v,z}$ is the column index of the component in row $(v+1)$, to which the component with coordinate (v,z) pairs.

Output:

$\boldsymbol{D}$: The optimal state of the reassembly inventory (the states with the highest total score), $\boldsymbol{D}=\{\boldsymbol{U}, \boldsymbol{B}\}$

W: The total score of the reassembly inventory

Local:

b, u, W', $\boldsymbol{U'}$, $\boldsymbol{B'}$, $\boldsymbol{D'}=\{\boldsymbol{U'}, \boldsymbol{B'}\}$

```
Assign U and B arbitrarily, $U'=U$, $B'=B$
$W$ := compute the total score of state D
for $i$ = 1 up to $m$-1 do
    for $j$ = 1 up to $n$ do
        for $k$ = 1 up to $n$ do
            if $j \neq U(i+1, k)$
                $B'(i,j) := k$, $U'(i+1,k) := j$
                $u := U(i+1,k)$, $b := B(i,j)$
                $B'(i,u) := b, U'(i+1,b) := u$
                $W'$ := compute the total score of state $D'$
                if $W' > W$
                    $B := B', U := U', W := W'$
                end if
            end if
        end for
    end for
end for
```

Figure 5.7 is an illustrative example that shows one iteration step of the RSIA at a given state. The reassembly inventory in this example consists of four components and four CSs. For ease of illustration, this example does not consider missing components or the uncertainty of inspection. Different colors in the reassembly inventory correspond to different reassembly chains, and cells with the same color belong to the same reassembly chain.

At time step t (with $W_n(s, a) = 26$), action $A_t = a$ makes the reassembly system change to state $S_t = s$, and then the agent can choose between four actions a'_1, a'_2, a'_3, and a'_4. As action a'_3 corresponds to a state with the highest total score (27), action a'_3 is selected, and the reassembly table transfers to state $S_{t+1} = S'$, with the new total score $Q_{k+1}(s, a) = 27$. $S_{t+1} = S'$ is the state for step $t+1$, which is the initial state for the next iteration. This method can now be used to identify the optimal reassembly strategy.

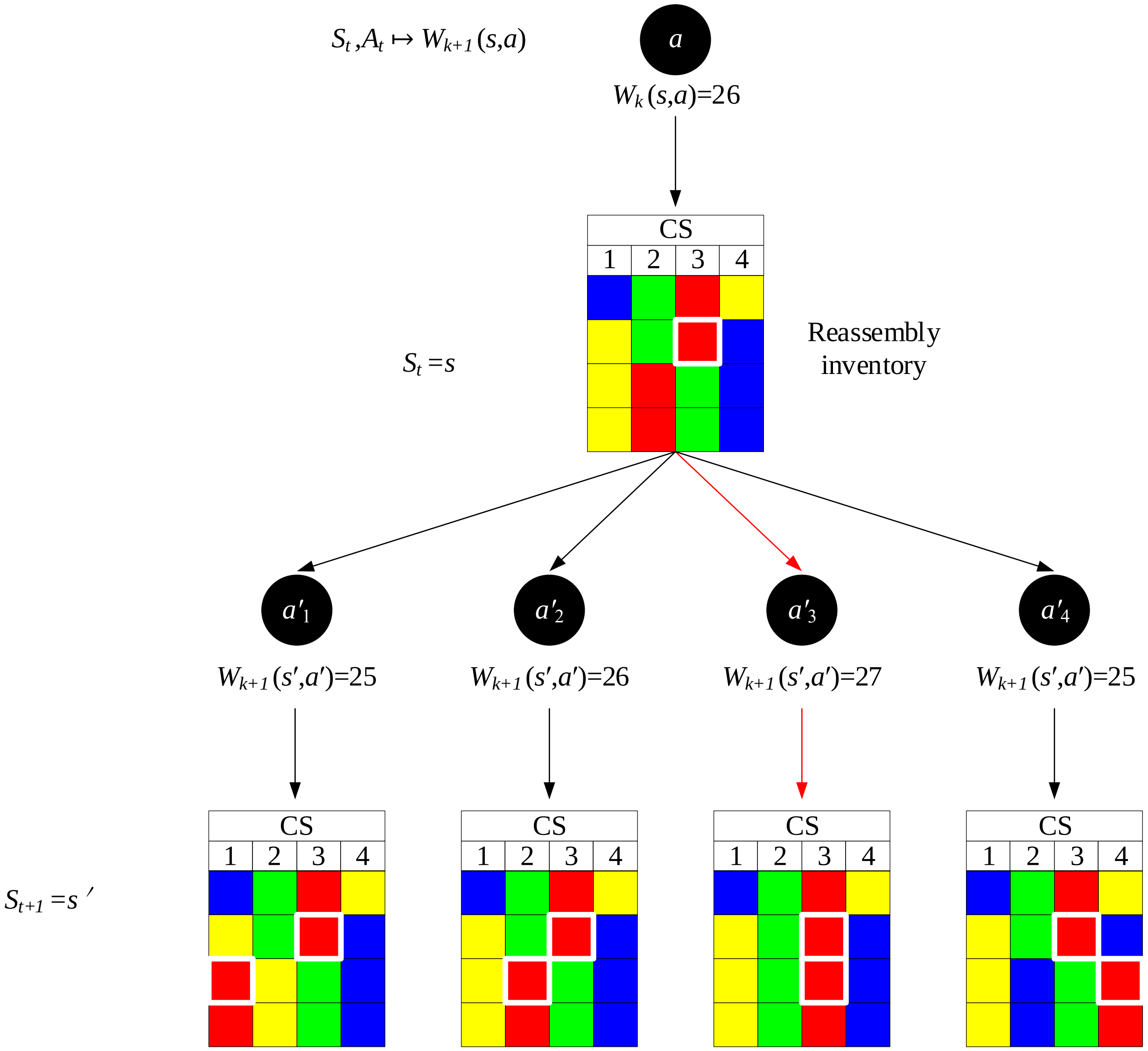


Figure 5.7. Backup diagram for iterating through the reassembly-value function. There are four components and four CSs. The diagram shows one step of the search process for the highlighted cell (the component in the second row, third column).

5.3 Case Study

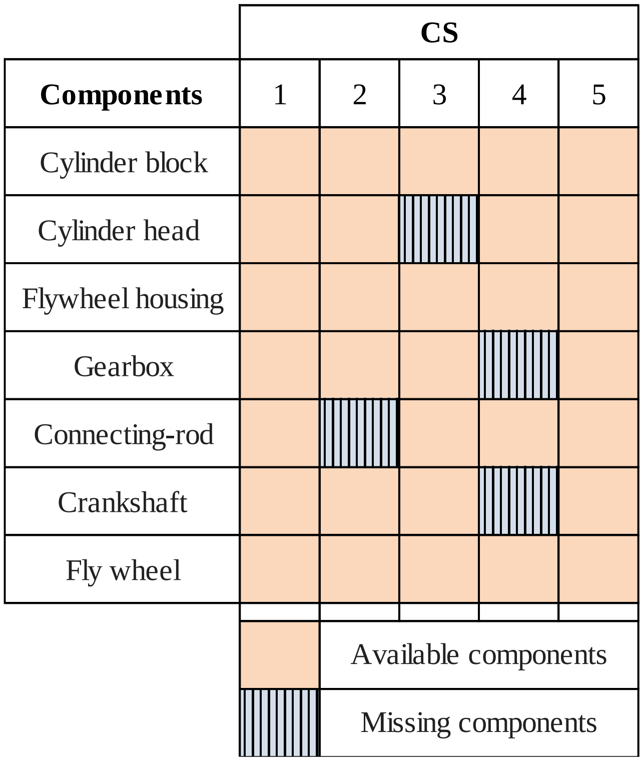
In this section, a numerical example using representative industry data from the remanufacturing of diesel engines is analyzed as a case study. To demonstrate the benefits of the component-based reassembly strategy, the scores of reassembled products using this strategy and a commonly used exhaustive reassembly strategy are compared. Two policies are used in the exhaustive reassembly strategy: i) returned components with differing quality/functional levels are not distinguished and are reassembled randomly, and ii) when any of the components in the reassembly inventory is missing, a corresponding new component will be purchased to complete a product.

Diesel engines consist of many complicated components and are widely remanufactured. Diesel engines are used in a wide range of equipment, and different customers have different performance requirements for each engine type. For instance, remanufactured diesel engines with lower quality levels satisfies the needs of customers of low-end equipment, because they prefer lower prices and may not need attributes such as high torque or high maximum power. The reassembly process of a certain model of diesel engines remanufactured by a Chinese company is analyzed. The company generally disassembles the diesel engine into seven components, which are then reassembled by the company. These components are the cylinder block, the cylinder head (set), the flywheel housing, the gearbox, the connecting-rod (set), the crankshaft, and the flywheel [140].

5.3.1 Validation of Component-oriented Optimal Reassembly Strategy

Ferguson et al. [70] showed that a remanufacturing system is most profitable when five quality levels are used. We consider five columns for the reassembly inventory. Each of the seven types of engine components from all the end-of-use engines was inspected and measured. Based on this measurement, each component was assigned a CS value. For ease of computation, the CSs were normalized to integer values from one to five (one corresponds to the lowest quality/functional level). The scored components were put into the reassembly inventory according to their normalized CSs. The probability of a measurement being placed into each category is estimated using a Monte Carlo simulation. Given a component with CS=c_i, a random score that follows a normal distribution, with $\mu=c_i$ and $\sigma=0.1$ is generated, this score is then categorized into the nearest integer, which is the categorized TS.

An example of the reassembly inventory for the engine component is shown in Figure 5.8. Because components have uncertain timing of return, for a reassembly inventory, any type of component with any score can be missing. In this illustration, four components (a cylinder head set with score 3, a gearbox with score 4, a connecting-rod set with score 2, and crankshaft with score 4) are missing. To consider this uncertainty, *N* reassembly inventory is randomly generated. For a reassembly inventory, each component has a probability $p=0.2$ to be missing. The average score of these *N* reassembly inventories is the score of a reassembly strategy.



	CS				
Components	1	2	3	4	5
Cylinder block	🟧	🟧	🟧	🟧	🟧
Cylinder head	🟧	🟧	▥	🟧	🟧
Flywheel housing	🟧	🟧	🟧	🟧	🟧
Gearbox	🟧	🟧	🟧	▥	🟧
Connecting-rod	🟧	▥	🟧	🟧	🟧
Crankshaft	🟧	🟧	🟧	▥	🟧
Fly wheel	🟧	🟧	🟧	🟧	🟧

🟧	Available components
▥	Missing components

Figure 5.8. Reassembly inventory for the seven pieces of engine components.

At the company where the data was collected, a remanufactured engine assembled using an exhaustive strategy is sold for 5,000 – 6,000 USD, which is 50-60% of the price of a new engine.

We assume that the selling price of reassembled engines is proportional to their scores, and that a willing customer exists for all reassembled engines (at a price proportional to the score). This is a common assumption in remanufacturing systems planning literature. The data from some companies (including Bosch, Cisco, and IBM) that perform remanufacturing show that the average demand for remanufactured products is (typically) much higher than the number of returned products in any given period except at the beginning of the product's lifecycle [70].

To quantitatively analyze the influence of using new components when components with certain scores are missing, Equation(5.11) is used to transform the price of new components to equivalent scores,

$$e_{i,j} = -\frac{g_{i,j}}{\sum_{i=1}^{7} g_{i,j}} \,, \tag{5.11}$$

in which, $g_{i,j}$ is the price of the new components, $i \in \{0,1,\ldots,7\}$, $j \in \{0,1,\ldots,5\}$. Remanufacturers can adjust Equation (5.11) if they are more (or less) sensitive to prices. The prices of new engine components and the corresponding transformed scores are shown in Table 5.3.

Table 5.3. Price and equivalent score of components in a diesel engine.

Component	Price, $g_{i,j}$ (USD)	Equivalent score, $e_{i,j}$, Equation (5.11)
cylinder block	1400	-0.42
cylinder head	820	-0.25
flywheel housing	110	-0.03
gearbox	110	-0.03
connecting-rod	230	-0.07
crankshaft	580	-0.17
flywheel	80	-0.02

The expected total score of the reassembly inventory $W(S_t,A_t)$ is computed using the objective function form one, and D_t is computed by Equation (5.12),

$$D_t = \sum (d_{h,k}^t + e_{i,j}^t)\,, \tag{5.12}$$

in which, $d^t_{h,k}$ is the inventory cost (component holding cost) for component $\{h,k\}$ in the reassembly inventory, if the component is not used in reassembly. t is the iteration index. When the score of a chain is negative because of using new components, components in this chain will not be assembled.

To estimate the number of reassembly inventory, N, for the average scores, $\overline{W}$, to converge to the expected score of a reassembly strategy, W, both our process and the exhaustive reassembly process are simulated from 2 to 30 times, and the values of $\overline{W}$ are recorded. The relationship between $\overline{W}$ and N is shown in Figure 5.9. The scores of the reassembly system controlled by the component-oriented reassembly strategy (CORS) and the exhaustive reassembly strategy converges to 14.05 and 10.04, respectively, after scores of 20 reassembly inventories are averaged. To balance the computational load and accuracy, the value of N is set to 20 in the sensitivity analysis of the following subsection.

As shown in Figure 5.9, the component-oriented reassembly model can greatly improve the remanufacturer’s ability to control the quality (or functional level) of a remanufactured engine, and thus can lead to increased profits.

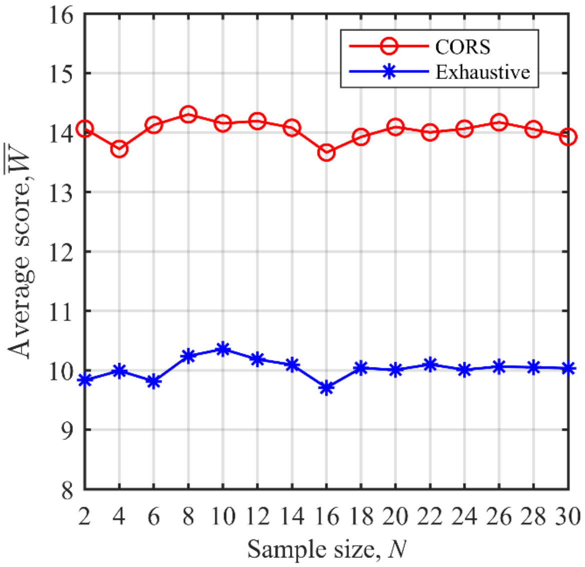
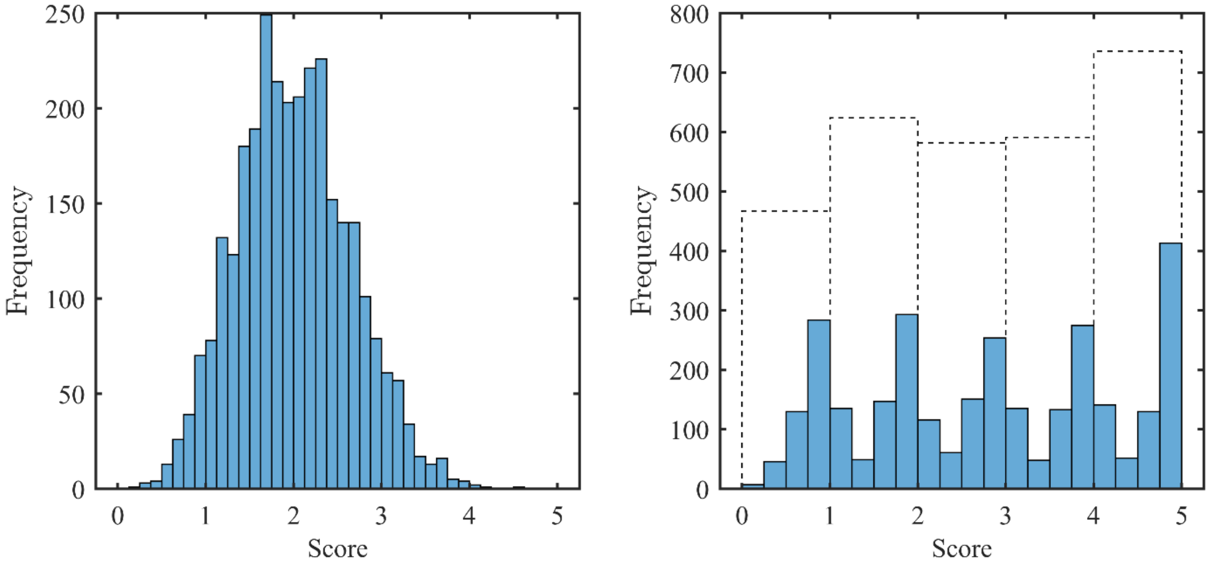


Figure 5.9. The relationship between *W* and *N* for *p*=0.2, *d*(*h*, *k*)=0.01.

The exhaustive reassembly strategy and the CORS assemble products with different distributions of total scores. Figure 5.10 shows the histograms of the total score using the exhaustive reassembly strategy and the CORS. In each of the figures, 3000 diesel engines are assembled. In Figure 5.10 (a), the distribution of the exhaustive reassembly strategy creates a roughly normal distribution of the total product scores, as would be expected from a random reassembly process. In Figure 5.10 (b) (dashed lines), the CORS creates total scores that span across the score range; there is a clustering of scores for each quality level. The purchasing of new components and the inventory costs decrease the score of the total products, which is shown by the fractional scores of products to the left of the values 1, 2, 3, 4, and 5 (solid lines in Figure 5.10 (b)). If the new component cost and inventory cost were zero, the four bars to the left of 1, 2, 3, 4 and 5 would all add up to the bars on the integers (dashed lines in Figure 5.10 (b)), and a relatively flat distribution of product scores would be seen. The wide distribution generated by the CORS is able to meet a wider range of customer needs, and also produces a significantly larger number of high quality products than the exhaustive reassembly strategy.



(a) exhaustive reassembly strategy

(b) component-oriented model

Figure 5.10. Histogram of total score for 600 simulations.

5.3.2 Sensitivity Analysis

The numerical case study described above demonstrate the advantages of the component-oriented reassembly model and the effectiveness of the methods proposed in this chapter. Some of the reassembly system parameters are selected subjectively, which may affect the experiment outcome. To study the relationship between the parameters and performance of the reassembly system, sensitivity analysis was carried out. Remanufacturers can adjust the parameters of their reassembly system based on the results of the sensitivity analysis. The averaged total score of the reassembly system is computed while one parameter is varied, and the other parameters are fixed at the values shown in Table 5.4.

Table 5.4. Fixed parameters for the sensitivity analysis.

Sample size, N	Missing probability, p	Equivalent score, $e_{i,j}$	Inventory cost, $d_{h,k}$	Standard deviation, σ
20	0.2	value in Table 5.3	0.01	0.1

- Probability of missing components, p.

  The probability of missing components is changed in the range of 0 to 1. As shown in Figure 5.11 (a), when the probability of missing components is high (for instance when the components are returned at random times during their useful life and the quantities of returned components are more uncertain), the total score of the reassembly system decreases for both the CORS and the exhaustive strategy, while the average total score of the component-oriented model is higher than the exhaustive strategy by about 40% for all probabilities of missing components.

- The equivalent score of the price for purchasing new components, $e_{i,j}$.

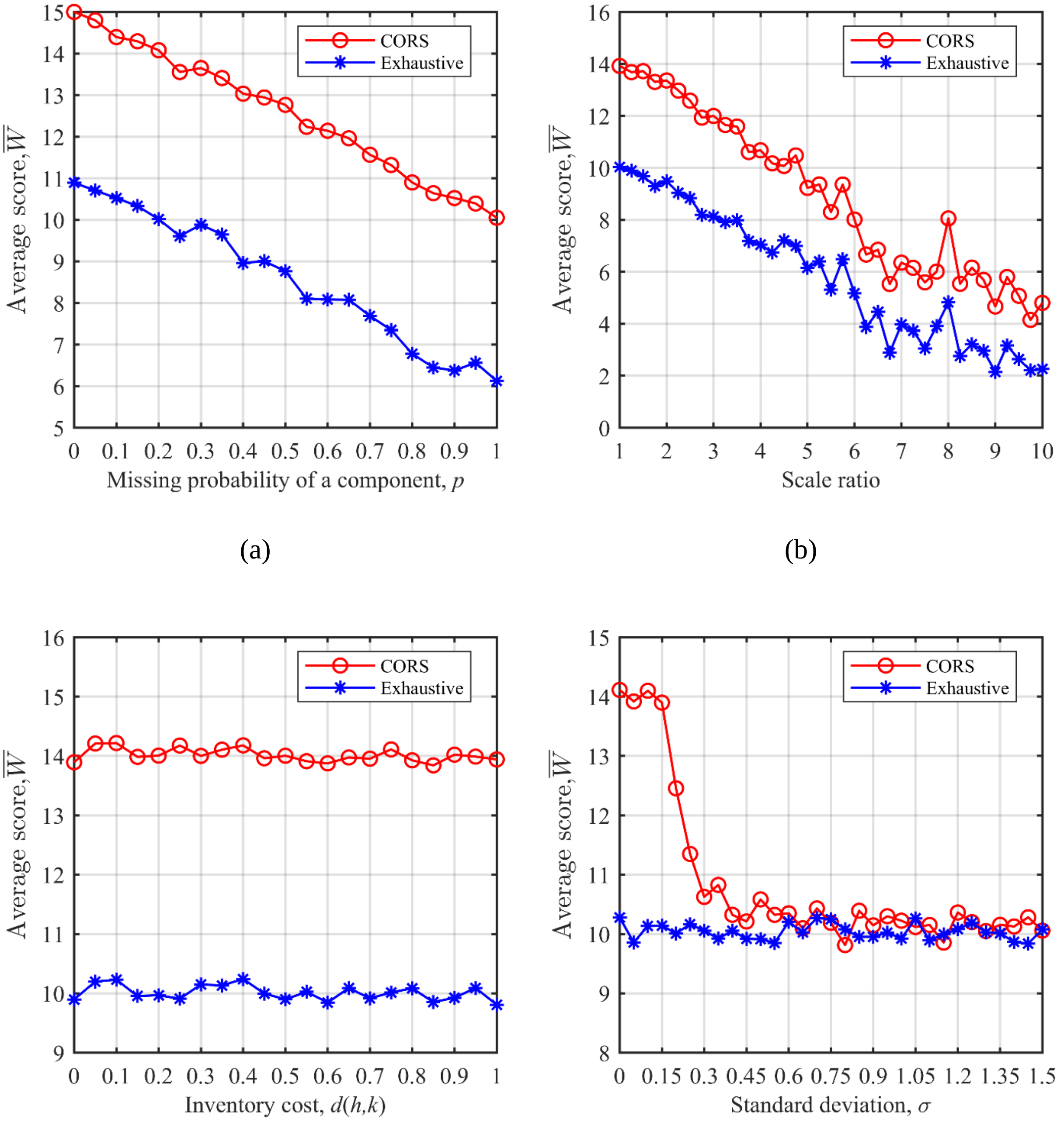
  The influence of the price of new components is studied by multiplying the equivalent Score, $e_{i,j}$, in Table 5.3 by scaling factors ranging from 1 to 10. As shown in Figure 5.11 (b), when the prices of the new components increase, the total score for both the CORS and the exhaustive reassembly strategy of the reassembly system decrease. If the reassembly system is controlled by the exhaustive strategy, the total score may be negative when the summation of the equivalent scores of the new components used in a reassembled product is higher than the total score of the product. This situation means that a reassembled product is not profitable. The CORS can avoid this situation by avoiding assembling those components.

- Inventory cost, $d_{h,k}$

Figure 5.11 (c) shows the influence of the cost of maintaining an inventory of unused components. Because the price of the new engine components are relatively low compared to the selling price of a remanufactured new engine (in the initial simulation, $e_{i,j}=1$), it is more profitable if new components are used to fill the gaps caused by missing components. This result shows that the inventory costs have little effect on the total scores of the products if the prices of new components are low compared to the selling price of the remanufactured products.

- Standard deviation of the true score, σ

  To study how the accuracy of the inspection process influences the CORS, a series of simulations are carried out with measurement uncertainty, σ, ranging from 0 to 1.5. As shown in Figure 5.11 (d), when the uncertainty of the inspection process is low, i.e., when σ is small (the CS is an accurate estimation of TS), the average total score of the CORS is higher than the exhaustive strategy by about 40%, but when σ is bigger than 0.6, the precision of the inspection processes is too low for the CORS to discriminate between scores with different TSs. Based on this comparison, the remanufacturers can estimate whether it would be cost-effective to inspect the returned components.



(a) (b)

(c) (d)

Figure 5.11. Results of sensitivity analysis.

The sensitivity analysis shows that the probability of missing components and the price of new components can cause the average score of the two strategies decrease, but the CORS performs 40% better than the exhaustive model on average. The inventory cost has few effects on the total score if the price of new components is relatively low compared to the selling price of a remanufactured product. The precision of the inspection process is a critical factor for the component-oriented reassembly process. The sensitivity analysis shows that the CORS performs at least as well, and sometimes significantly better than, the exhaustive strategy.

5.4 Conclusions and future research

A component-oriented model was designed to control the reassembly process of remanufacturing systems by taking into consideration the uncertainties of returned components and a variety of customer needs. In the component-oriented model, the returned components are inspected, and a score is assigned to each component according to its quality/functional level. The components are paired with one-another and chains of component pairs are used to assemble and evaluate products based on product scores, under the control of a reassembly strategy. These reassembly strategies are evaluated, and the product scores are calculated using different objective functions, which represent different goals or remanufacturing scenarios. The reassembly strategies are evaluated using an agent-environment system, which is modelled as a Markov decision process (MDP). The optimal reassembly strategy is identified using a reassembly-score iteration algorithm (RSIA), and the performance of the RSIA is evaluated by comparison with a commonly used exhaustive (random) reassembly strategy in a diesel engine case study. The case study showed that compared with the exhaustive reassembly strategy, the component-oriented reassembly method improved the performance of the reassembly system up to 40%, created a distribution of products that can meet a wider variety of customer needs, and created more high-quality products. Sensitivity analysis showed that variables such as the probability of missing components, the price of new components, and the accuracy of the inspection process affects the performance of the models.

There are several directions for future research. The optimal reassembly strategy is searched based on a static reassembly inventory, in which the state of the inventory is fully observable and does not change during the control process. Future extensions to this work can consider a dynamic inventory, in which the state is not fully observable or is unstable. In this research, two forms of the objective function are provided, but other forms of the objective function could be used to consider other objectives such as optimization of the reassembly strategies for environmental performance. Additional simulation could be carried out to study the influence of inspection uncertainty.

6. CONCLUSION

In this dissertation, the issue of product quality control in manufacturing has been addressed. To start, the importance of having an overall consideration of multiple stages of the product life cycle in quality control was analyzed. This means optimally managing product quality requires finding tradeoffs among multiple relations surrounding a product. It was also pointed out that components are the bridges connecting these relations. A series of studies were carried out to get a fundamental understanding of the relationship between component and product quality, with an emphasis on improving the competitiveness of a manufacturer, i.e., to improve quality, reduce cost, and limit environmental impact. Contributions of the dissertation are concluded below.

First, a tolerance allocation method that minimizes cost by jointly considering process variation and tolerance specifications was developed. This model employed a cost model that includes processing cost, scrap cost, and quality loss (impact of quality deviation evaluated in a monetary scale). The relation between manufacturer, user, design, and processing were embedded in the cost model. The results showed that the proposed model decreased the average cost by avoiding unnecessary process precision, more effectively allocating tolerances among individual components, and optimizing specification limits of components. The proposed method approached the tolerance allocation problem by optimizing both the precision of manufacturing processes and the specification limits of components. This model established a connection between product design and product manufacturing. As was evident from the case study, the proposed approach provided a superior solution to the tolerance allocation problem.

Then, a second tolerance allocation model that considered both product design (tolerance selection) and operation planning (or production rate selection) was developed. Relations among production rate, production cost, processing precision, and waste were considered. This model solved the tolerance problem from the root cause, i.e., the variation in production. Compared to earlier models, this method produced more satisfactory products at a lower cost while producing less waste. Based on the results, some managerial insights were given. For example, it was found that when the precision of a process was high, it was not necessary from an economic standpoint to inspect the quality of individual components. For poor precision processes, inspecting the quality of individual components was the preferred approach from a cost/throughput standpoint.

To extend traditional statistical process control to the area of processes monitoring where multiple sensors are deployed, a model to detect anomalies in time series data with long length and high dimensionality was developed. This model was based on recurrent neural networks, and the parameters of the neural networks can be trained using data acquired during routine system operation. The model takes time series data as an input and reconstructs the input data. Time series data with an anomaly causes patterns in the reconstruction errors that are inconsistent with error patterns of anomaly-free data. The performance of the proposed method was assessed using data from a diesel engine assembly process. Three common types of anomalies were detected from the time series data.

Quality control of remanufacturing was also studied. A component-oriented reassembly model was proposed. In this model, returned components were inspected and assigned scores according to their quality/function, and categorized in a reassembly inventory. Based on the reassembly inventory, components were paired under the control of a reassembly strategy. To evaluate the performance of different reassembly strategies under uncertain conditions, the reassembly problem was described by an agent-environment system, and the platform was modeled as a Markov decision process. A reassembly-score iteration algorithm was developed to identify the optimal reassembly strategy. The effectiveness of the method was demonstrated via a case study using the reassembly process of diesel engines.

In summary, this dissertation has presented a series of quality management models. These models are based on studying the relationship between component and product quality at multiple stages of a product life cycle. It was found that effectively managing this relationship is fundamental of improving product quality, saving cost, and reducing environmental impact. We believe that further research on quality control will help improve the competitiveness of manufacturers greatly.

REFERENCES

[1] J. M. Juran and A. D. F. Joseph, *Juran's Quality Handbook*, vol. 1, no. 3. McGraw-Hill Companies, Inc., 2010.

[2] R. DeVor, T. Chang, and J. W. Sutherland, *Statistical Quality Design and Control*, 2nd ed. Pearson Prentice Hall, 2007.

[3] Y. Wang, S. Calhoun, L. Bosman, and J. W. Sutherland, "Tolerance allocations on products: A life cycle engineering perspective," *Procedia CIRP*, vol. 80, pp. 174–179, 2019.

[4] Z. Dong and W. Hu, "Optimal process sequence identification and optimal process tolerance assignment in computer-aided process planning," *Comput. Ind.*, vol. 17, no. 1, pp. 19–32, 1991.

[5] K. Chase, W. H. Greenwood, B. G. Loosli, and L. F. Hauglund, "Least cost tolerance allocation for mechanical assemblies with automated process selection," *Manuf. Rev.*, pp. 49–59, 1990.

[6] K. W. Chase and W. H. Greenwood, "Design issues in mechanical tolerance analysis," *Manufacturing Review*, vol. 1, no. 1. pp. 50–59, 1988.

[7] A. Sanz-Lobera, M. A. Sebastián, and J. M. Pérez, "New cost-tolerance model for mechanical part design," *Int. J. Adv. Manuf. Technol.*, vol. 51, no. 5–8, pp. 421–430, 2010.

[8] F. H. Speckhart, "Calculation of tolerance based on a minimum cost approach," *J. Manuf. Sci. Eng.*, vol. 94, no. 2, pp. 447–453, 1972.

[9] G. H. Sutherland and B. Roth, "Mechanism design: accounting for manufacturing tolerances and costs in function generating problems," *J. Manuf. Sci. Eng.*, vol. 97, no. 1, pp. 283–286, 1975.

[10] C. H. Chen and H. S. Kao, “The determination of optimum process mean and screening limits based on quality loss function,” *Expert Syst. Appl.*, vol. 36, no. 3 PART 2, pp. 7332–7335, 2009.

[11] J. R. He, “Tolerancing for manufacturing via cost minimization,” *Int. J. Mach. Tools Manuf.*, vol. 31, no. 4, pp. 455–470, Jan. 1991.

[12] H.-G. R. Choi, M. Park, and E. Salisbury, “Optimal Tolerance Allocation With Loss Functions,” *J. Manuf. Sci. Eng.*, vol. 122, no. 3, p. 529, 2000.

[13] K. Geetha, D. Ravindran, M. S. Kumar, and M. N. Islam, “Concurrent tolerance allocation and scheduling for complex assemblies,” *Robot. Comput. Integr. Manuf.*, vol. 35, pp. 84–95, Oct. 2015.

[14] H. Vasseur, T. R. Kurfess, and J. Cagan, “Use of a quality loss function to select statistical tolerances,” *J. Manuf. Sci. Eng. Trans. ASME*, vol. 119, no. 3, pp. 410–416, 1997.

[15] P. Wang and M. Liang, “An integrated approach to tolerance synthesis, process selection and machining parameter optimization problems,” *Int. J. Prod. Res.*, vol. 43, no. 11, pp. 2237–2262, 2005.

[16] S. Liu and J. Qiu, “Tolerance design considering the resources and environment characters during manufacturing process,” *Comput. Integr. Manuf. Syst.*, vol. 7, 2011.

[17] M. P. Sealy, Z. Y. Liu, D. Zhang, Y. B. Guo, and Z. Q. Liu, “Energy consumption and modeling in precision hard milling,” *J. Clean. Prod.*, vol. 135, pp. 1591–1601, 2016.

[18] J. Yan and L. Li, “Multi-objective optimization of milling parameters-the trade-offs between energy, production rate and cutting quality,” *J. Clean. Prod.*, vol. 52, pp. 462–471, 2013.

[19] W. H. Greenwood and K. W. Chase, “Worst case tolerance analysis with nonlinear problems,” *J. Manuf. Sci. Eng.*, vol. 110, no. 3, pp. 232–235, 1988.

[20] P. K. Singh, P. K. Jain, and S. C. Jain, *A genetic algorithm-based solution to optimal tolerance synthesis of mechanical assemblies with alternative manufacturing processes: Focus on complex tolerancing problems*, vol. 42, no. 24. 2004.

[21] A. N. Haq, K. Sivakumar, R. Saravanan, and V. Muthiah, “Tolerance design optimization of machine elements using genetic algorithm,” *Int. J. Adv. Manuf. Technol.*, vol. 25, no. 3–4, pp. 385–391, 2005.

[22] E. Zahara and Y. T. Kao, “A hybridized approach to optimal tolerance synthesis of clutch assembly,” *Int. J. Adv. Manuf. Technol.*, vol. 40, no. 11–12, pp. 1118–1124, 2009.

[23] Y. Zhang, G. Zhang, T. Qu, Y. Liu, and R. Y. Zhong, “Analytical target cascading for optimal configuration of cloud manufacturing services,” *J. Clean. Prod.*, vol. 151, pp. 330–343, 2017.

[24] L. R. Kumar, K. Padmanaban, and C. Balamurugan, “Least cost-tolerance allocation based on Lagrange multiplier,” *Concurr. Eng. Res. Appl.*, vol. 24, no. 2, pp. 164–177, 2016.

[25] M. Siva Kumar and B. Stalin, “Optimum tolerance synthesis for complex assembly with alternative process selection using Lagrange multiplier method,” *Int. J. Adv. Manuf. Technol.*, vol. 44, no. 3–4, pp. 405–411, 2009.

[26] M. Tlija, M. Ghali, and N. Aifaoui, “Integrated CAD tolerancing model based on difficulty coefficient evaluation and Lagrange multiplier,” *Int. J. Adv. Manuf. Technol.*, vol. 101, no. 9–12, pp. 2519–2532, 2019.

[27] S. Shin, P. Kongsuwon, and B. R. Cho, “Development of the parametric tolerance modeling and optimization schemes and cost-effective solutions,” *Eur. J. Oper. Res.*, vol. 207, no. 3, pp. 1728–1741, Dec. 2010.

[28] A. Sofiana, C. N. Rosyidi, and E. Pujiyanto, “Product quality improvement model considering quality investment in rework policies and supply chain profit sharing,” *J. Ind. Eng. Int.*, vol. 15, no. 4, pp. 637–649, 2019.

[29] A. J. Qureshi, J.-Y. Dantan, V. Sabri, P. Beaucaire, and N. Gayton, “A statistical tolerance analysis approach for over-constrained mechanism based on optimization and Monte Carlo simulation,” *Comput. Des.*, vol. 44, no. 2, pp. 132–142, Feb. 2012.

[30] F. Wu, J.-Y. Dantan, A. Etienne, A. Siadat, and P. Martin, “Improved algorithm for tolerance allocation based on Monte Carlo simulation and discrete optimization,” *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1402–1413, May 2009.

[31] S. Hoffenson, A. Dagman, and R. Söderberg, “Visual quality and sustainability considerations in tolerance optimization: A market-based approach,” *Int. J. Prod. Econ.*, vol. 168, pp. 167–180, 2015.

[32] A. Haghighi and L. Li, “Joint Asymmetric Tolerance Design and Manufacturing Decision-Making for Additive Manufacturing Processes,” *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1259–1270, 2019.

[33] W. Huang, T. Phoomboplab, and D. Ceglarek, “Process capability surrogate model-based tolerance synthesis for multi-station manufacturing systems,” *IIE Trans. (Institute Ind. Eng.*, vol. 41, no. 4, pp. 309–322, 2009.

[34] C. N. Rosyidi, R. Murtisari, and W. Jauhari, “A concurrent optimization model for suppliers selection, tolerance and component allocation with fuzzy quality loss,” *Cogent Eng.*, vol. 3, no. 1, 2016.

[35] C. N. Rosyidi, R. Murtisari, and W. A. Jauhari, “A concurrent optimization model for supplier selection with fuzzy quality loss,” *J. Ind. Eng. Manag.*, vol. 10, no. 1, pp. 98–110, 2017.

[36] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, “Deep learning for smart manufacturing: Methods and applications,” *J. Manuf. Syst.*, vol. 48, pp. 144–156, 2018.

[37] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, “Intelligent Manufacturing in the Context of Industry 4.0: A Review,” *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[38] G. Adamson, L. Wang, M. Holm, and P. Moore, “Cloud manufacturing – a critical review of recent development and future trends,” *Int. J. Comput. Integr. Manuf.*, no. March, pp. 347–380, 2015.

[39] W. J. Lee, K. Xia, N. L. Denton, B. Ribeiro, and J. W. Sutherland, “Development of a speed invariant deep learning model with application to condition monitoring of rotating machinery,” *J. Intell. Manuf.*, pp. 1–14, 2020.

[40] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. G. Jun, and J. W. Sutherland, “Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data,” *Procedia CIRP*, vol. 80, pp. 506–511, 2019.

[41] R. Chalapathy and S. Chawla, “Deep Learning for Anomaly Detection: A Survey,” *arXiv Prepr.*, 2019.

[42] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, “Towards Parameter-Free Data Mining,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 206–215.

[43] E. Keogh, J. Lin, and A. Fu, “HOT SAX Finding the Most Unusual Time Series Subsequence: Algorithms and Applications,” in *Fifth IEEE International Conference on Data Mining (ICDM’05)*, 2005, p. 8.

[44] X. Wang, J. Lin, N. Patel, and M. Braun, “A self-learning and online algorithm for time series anomaly detection, with application in CPU manufacturing,” in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1823–1832.

[45] S. H. Seifi, W. Tian, H. Doude, M. A. Tschopp, and L. Bian, “Layer-Wise Modeling and Anomaly Detection for Laser-Based Additive Manufacturing,” *J. Manuf. Sci. Eng. Trans. ASME*, vol. 141, no. 8, p. 081013, 2019.

[46] L. Chen, G. Xu, S. Zhang, W. Yan, and Q. Wu, “Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks,” *J. Manuf. Syst.*, vol. 54, no. June 2019, pp. 1–11, 2020.

[47] D. Xiao, Y. Huang, X. Zhang, H. Shi, C. Liu, and Y. Li, “Fault Diagnosis of Asynchronous Motors Based on LSTM Neural Network,” *Proc. - 2018 Progn. Syst. Heal. Manag. Conf. PHM-Chongqing 2018*, no. 2017, pp. 540–545, 2019.

[48] W. J. Lee, H. Wu, A. Huang, and J. W. Sutherland, “Learning via acceleration spectrograms of a DC motor system with application to condition monitoring,” *Int. J. Adv. Manuf. Technol.*, vol. 106, no. 3–4, pp. 803–816, 2020.

[49] L. Bontemps, V. L. Cao, J. McDermott, and N.-A. Le-Khac, “Collective Anomaly Detection based on Long Short Term Memory Recurrent Neural Network,” in *International Conference on Future Data and Security Engineering*, 2016, pp. 141–152.

[50] J. Sipple, “Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure,” 2020.

[51] C. Zhang *et al.*, “A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 1409–1416.

[52] P. Wang, Ananya, R. Yan, and R. X. Gao, “Virtualization and deep recognition for system fault classification,” *J. Manuf. Syst.*, vol. 44, pp. 310–316, 2017.

[53] P. Malhotra, L. Vig, G. Shrof, and P. Agarwal, “Long short term memory networks for anomaly detection in time series,” in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015, no. April, pp. 22–24.

[54] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 387–395, 2018.

[55] P. Hayton, S. Utete, D. King, S. King, P. Anuzis, and L. Tarassenko, “Static and dynamic novelty detection methods for jet engine health monitoring,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 365, no. 1851, pp. 493–514, 2007.

[56] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, “LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection,” in *arXiv preprint*, 2016.

[57] P. Malhotra, T. V. Vishnu, L. Vig, P. Agarwal, and G. Shroff, “TimeNet: Pre-trained deep recurrent neural network for time series classification,” in *arXiv preprint arXiv*, 2017.

[58] Y.-H. Yoo, U.-H. Kim, and J.-H. Kim, “Recurrent Reconstructive Network for Sequential Anomaly Detection,” *IEEE Trans. Cybern.*, pp. 1–12, 2019.

[59] Y. Tan *et al.*, “An encoder-decoder based approach for anomaly detection with application in additive manufacturing,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1008–1015.

[60] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *arXiv preprint*, 2014, vol. 4, no. January.

[61] D. T. Shipmon, J. M. Gurevitch, P. M. Piselli, and S. Edwards, *Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data*. 2017.

[62] M. Liu, Q. Ke, S. Song, and X. Zhou, “Active Remanufacturing Timing Determination Based on Failure State Assessment,” in *Re-engineering Manufacturing for Sustainability*, Singapore: Springer, Singapore, 2013, pp. 615–619.

[63] S. Song, M. Liu, Q. Ke, and H. Huang, “Proactive remanufacturing timing determination method based on residual strength,” *Int. J. Prod. Res.*, vol. 53, no. 17, pp. 5193–5206, 2015.

[64] C. Su and A. Xu, “Buffer allocation for hybrid manufacturing/remanufacturing system considering quality grading,” *Int. J. Prod. Res.*, vol. 52, no. 5, pp. 1269–1284, 2014.

[65] T. Sakai and S. Takata, “Reconfiguration management of remanufactured products for responding to varied user needs,” *CIRP Ann.*, vol. 61, no. 1, pp. 21–26, Jan. 2012.

[66] M. R. Galbreth and J. D. Blackburn, “Optimal acquisition and sorting policies for remanufacturing,” *Prod. Oper. Mangement*, vol. 15, no. 3, pp. 384–392, 2006.

[67] T. Tolio *et al.*, “Design, management and control of demanufacturing and remanufacturing systems,” *CIRP Ann. - Manuf. Technol.*, vol. 66, no. 2, pp. 585–609, Jan. 2017.

[68] C. Zikopoulos and G. Tagaras, “On the attractiveness of sorting before disassembly in remanufacturing,” *IIE Trans.*, vol. 40, no. 3, pp. 313–323, 2008.

[69] X. Jin, J. Ni, and Y. Koren, “Optimal control of reassembly with variable quality returns in a product remanufacturing system,” *CIRP Ann. - Manuf. Technol.*, vol. 60, no. 1, pp. 25–28, 2011.

[70] M. Ferguson, V. D. J. Guide, E. Koca, and G. C. Souza, “The value of quality grading in remanufacturing,” *Prod. Oper. Manag.*, vol. 18, no. 3, pp. 300–314, 2009.

[71] “Recycling of highly functional components,” 2018. [Online]. Available: http://www.ricoh.com/environment/product/resource/01_01.html. [Accessed: 14-Apr-2018].

[72] M. Denizel, M. Ferguson, and G. “Gil” C. Souza, “Multiperiod remanufacturing planning with uncertain quality of inputs,” *IEEE Trans. Eng. Manag.*, vol. 57, no. 3, pp. 394–404, 2010.

[73] S. Behdad, A. S. Williams, and D. Thurston, “End-of-Life Decision Making With Uncertain Product Return Quantity,” *J. Mech. Des.*, vol. 134, no. 10, p. 100902, 2012.

[74] N. Aras, T. Boyaci, and V. Verter, “The effect of categorizing returned products in remanufacturing,” *IIE Trans. (Institute Ind. Eng.*, vol. 36, no. 4, pp. 319–331, 2004.

[75] A. R. Mashhadi, B. Esmaeilian, and S. Behdad, “Uncertainty Management in Remanufacturing Decisions: A Consideration of Uncertainties in Market Demand, Quantity, and Quality of Returns,” *ASCE-ASME J. Risk Uncert. Engrg. Sys., Part B Mech. Engrg.*, vol. 1, no. 2, p. 021007, 2015.

[76] M. Liu, C. Liu, and Q. Zhu, “Optional classification for reassembly methods with different precision remanufactured parts,” *Assem. Autom.*, vol. 34, no. 4, pp. 315–322, 2014.

[77] W. Shen, K. Pang, C. Liu, M. Ge, Y. Zhang, and X. Wang, “The quality control method for remanufacturing assembly based on the Jacobian-torsor model,” *Int. J. Adv. Manuf. Technol.*, vol. 81, no. 1–4, pp. 253–261, 2015.

[78] A. R. Mashhadi, S. Behdad, and J. Zhuang, “Agent Based Simulation Optimization of Waste Electrical and Electronics Equipment Recovery,” *J. Manuf. Sci. Eng.*, vol. 138, no. 10, p. 101007, 2016.

[79] X. Zhang, H. Zhang, Z. Jiang, and Y. Wang, “A decision-making approach for end-of-life strategies selection of used parts,” *Int. J. Adv. Manuf. Technol.*, vol. 87, no. 5–8, pp. 1457–1464, 2016.

[80] X. Jin, S. J. Hu, J. Ni, and G. Xiao, “Assembly strategies for remanufacturing systems with variable quality returns,” *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 76–85, 2013.

[81] O. Mont, C. Dalhammar, and N. Jacobsson, “A new business model for baby prams based on leasing and product remanufacturing,” *J. Clean. Prod.*, vol. 14, no. 17, pp. 1509–1518, 2006.

[82] E. Morse *et al.*, “Tolerancing: Managing uncertainty from conceptual design to final product,” *CIRP Ann.*, vol. 67, no. 2, pp. 695–717, 2018.

[83] P. K. Sing, S. C. Jain, and P. K. Jain, “Comparative study of genetic algorithm and simulated annealing for optimal tolerance design formulated with discrete and continuous variables,” *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 219, no. 10, pp. 735–758, 2005.

[84] L. Laperrière and H. A. Elmaraghy, “Tolerance analysis and synthesis using Jacobian transforms,” *CIRP Ann. - Manuf. Technol.*, vol. 49, no. 1, pp. 359–362, 2000.

[85] R. Söderberg, K. Wärmefjord, and L. Lindkvist, “Variation simulation of stress during assembly of composite parts,” *CIRP Ann. - Manuf. Technol.*, vol. 64, no. 1, pp. 17–20, 2015.

[86] G. Taguchi, *Introduction to Quality Engineering*. Tokyo: Asian Productivity Organization, 1986.

[87] R. E. Devor, T. Chang, and J. W. Sutherland, *Statistical Quality Design and Control*, 2nd ed. New Jersey: Prentice Hall, 2007.

[88] A. Haghighi and L. Li, “Study of the relationship between dimensional performance and manufacturing cost in fused deposition modeling,” *Rapid Prototyp. J.*, vol. 24, no. 2, pp. 395–408, 2018.

[89] B. Sarkar and S. Saren, “Product inspection policy for an imperfect production system with inspection errors and warranty cost,” *Eur. J. Oper. Res.*, vol. 248, no. 1, pp. 263–271, 2016.

[90] L. C. Leung and Y. V. Hui, “Dynamic management of cutting tools for flexible and quality machining,” *Int. J. Prod. Res.*, vol. 38, no. 14, pp. 3385–3401, 2000.

[91] H. C. Zhang and M. E. Huq, “Tolerancing techniques: The state-of-the-art,” *Int. J. Prod. Res.*, vol. 30, no. 9, pp. 2111–2135, 1992.

[92] A. A. Taleizadeh, M. S. Moshtagh, and I. Moon, “Pricing, product quality, and collection optimization in a decentralized closed-loop supply chain with different channel structures: Game theoretical approach,” *J. Clean. Prod.*, vol. 189, pp. 406–431, 2018.

[93] O. K. Mont, “Clarifying the concept of product–service system,” *J. Clean. Prod.*, vol. 10, pp. 237–245, 2002.

[94] J. M. Wilson, C. Piya, Y. C. Shin, F. Zhao, and K. Ramani, “Remanufacturing of turbine blades by laser direct deposition with its energy and environmental impact analysis,” *J. Clean. Prod.*, vol. 80, pp. 170–178, 2014.

[95] R. K. Bhushan, "Optimization of cutting parameters for minimizing power consumption and maximizing tool life during machining of Al alloy SiC particle composites," *J. Clean. Prod.*, vol. 39, pp. 242–254, 2013.

[96] H. A. Hegab, B. Darras, and H. A. Kishawy, "Towards sustainability assessment of machining processes," *J. Clean. Prod.*, vol. 170, pp. 694–703, 2018.

[97] F. Scholz, "Tolerance Stack Analysis Methods A Critical Review," 1995.

[98] Cheng-Jung Lin and Hsu-Pin Wang, "Optimal operation planning and sequencing: Minimization of tool changeovers," *Int. J. Prod. Res.*, vol. 31, no. 2, pp. 311–324, 1993.

[99] K. C. Kapur and B. R. Cho, "Economic design and development of specifications," *Qual. Eng.*, vol. 6, no. 3, pp. 401–417, 1994.

[100] T. Kim and C. H. Glock, "Production planning for a two-stage production system with multiple parallel machines and variable production rates," *Int. J. Prod. Econ.*, vol. 196, no. April 2016, pp. 284–292, 2018.

[101] G. Gazzola *et al.*, "Integrated Variable Importance Assessment in Multi-Stage Processes," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 3, pp. 343–355, 2018.

[102] C. H. Glock, "Batch sizing with controllable production rates," *Int. J. Prod. Res.*, vol. 48, no. 20, pp. 5925–5942, 2010.

[103] M. Khouja and A. Mehrez, "Economic Production Lot Size Model with Variable Production Rate and Imperfect Quality," *J. Oper. Res. Soc.*, vol. 45, no. 12, pp. 1405–1417, 1994.

[104] P. Nicolaou, D. L. Thurston, and J. V. Carnahan, "Machining quality and cost: Estimation and tradeoffs," *J. Manuf. Sci. Eng. Trans. ASME*, vol. 124, no. 4, pp. 840–851, 2002.

[105] E. Budak, “Analytical models for high performance milling. Part I: Cutting forces, structural deformations and tolerance integrity,” *Int. J. Mach. Tools Manuf.*, vol. 46, no. 12–13, pp. 1478–1488, 2006.

[106] W. Fan, C. H. Lee, and J. H. Chen, “A realtime curvature-smooth interpolation scheme and motion planning for CNC machining of short line segments,” *Int. J. Mach. Tools Manuf.*, vol. 96, pp. 27–46, 2015.

[107] Y. Wang, C. Yan, J. Yang, and C.-H. Lee, “Tool path generation algorithm based on covariant field theory and cost functional optimization and its applications in blade machining,” *Int. J. Adv. Manuf. Technol.*, vol. 90, no. 1–4, pp. 927–943, Apr. 2017.

[108] M. A. Farooq, R. Kirchain, H. Novoa, and A. Araujo, “Cost of quality: Evaluating cost-quality trade-offs for inspection strategies of manufacturing processes,” *Int. J. Prod. Econ.*, vol. 188, no. November 2016, pp. 156–166, 2017.

[109] A. Otsuka and F. Nagata, “Design method of Cpm-index based on product performance and manufacturing cost,” *Comput. Ind. Eng.*, vol. 113, pp. 921–927, Nov. 2017.

[110] R. Kawlra and W. Hancock, “Tolerance allocation methodology for manufacturing,” *SAE Tech. Pap.*, vol. 105, no. 1996, pp. 912–919, 1996.

[111] K. Svanberg, “Methods Based on Conservative Convex Separable,” *SIAM J. Optim.*, vol. 12, no. 2, pp. 555–573, 2002.

[112] C. X. Feng and A. Kusiak, “Robust tolerance design with the integer programming approach,” *J. Manuf. Sci. Eng. Trans. ASME*, vol. 119, no. 4, pp. 603–610, 1997.

[113] A. N. Haq and K. S. R. Saravanan, “Particle swarm optimization (PSO) algorithm for optimal machining allocation,” pp. 865–869, 2006.

[114] E. Budak and Y. Altintas, “Modeling and avoidance of static form errors in peripheral milling of plates,” *Int. J. Mach. Tools Manuf.*, vol. 35, no. 3, pp. 459–476, 1995.

[115] S. S. Yeh and P. L. Hsu, “Adaptive-feedrate interpolation for parametric curves with a confined chord error,” *CAD Comput. Aided Des.*, vol. 34, no. 3, pp. 229–237, 2002.

[116] G. Boothroyd and W. A. Knight, *Fundamentals of Metal Machining*, 2nd ed. New York: Marcel Dekker, 1989.

[117] E. M. Lim and C.-H. Meng, “Integrated planning for precision machining of complex surfaces. Part 1: cutting-path and feedrate optimization,” *Int. J. Mach. Tools Manuf.*, vol. 37, no. I, pp. 61–75, 1997.

[118] “NLopt,” 2020. [Online]. Available: https://nlopt.readthedocs.io/en/latest/.

[119] V. K. Omachonu, S. Suthummanon, and N. G. Einspruch, “The relationship between quality and quality cost for a manufacturing company,” *Int. J. Qual. Reliab. Manag.*, vol. 21, no. 3, pp. 277–290, 2004.

[120] E. Quatrini, F. Costantino, G. Di Gravio, and R. Patriarca, “Machine learning for anomaly detection and process phase classification to improve safety and maintenance activities,” *J. Manuf. Syst.*, vol. 56, no. November 2019, pp. 117–132, 2020.

[121] G. Aydemir and B. Acar, “Anomaly monitoring improves remaining useful life estimation of industrial machinery,” *J. Manuf. Syst.*, vol. 56, no. June, pp. 463–469, 2020.

[122] M. W. Milo, M. Roan, and B. Harris, “A new statistical approach to automated quality control in manufacturing processes,” *J. Manuf. Syst.*, vol. 36, pp. 159–167, 2015.

[123] D. Wulsin, J. Blanco, R. Mani, and B. Litt, “Semi-supervised anomaly detection for EEG waveforms using deep belief nets,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 375–395.

[124] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *ICLR 2015*, 2015, pp. 1–15.

[125] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[126] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[127] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1724–1734, 2014.

[128] K. Ortegon, L. Nies, and J. W. Sutherland, “Remanufacturing,” *CIRP Encyclopedia of Production Engineering.* Springer, Berlin, Heidelberg, p. 1044, 2014.

[129] P. Goodall, E. Rosamond, and J. Harding, “A review of the state of the art in tools and techniques used to evaluate remanufacturing feasibility,” *J. Clean. Prod.*, vol. 81, pp. 1–15, 2014.

[130] X. S. Si, W. Wang, C. H. Hu, M. Y. Chen, and D. H. Zhou, “A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation,” *Mech. Syst. Signal Process.*, vol. 35, no. 1–2, pp. 219–237, 2013.

[131] J. Ma and H. M. Kim, “ASME_JMSE_EXAMPLE,” *J. Mech. Des.*, vol. 136, no. 6, p. 061002, 2014.

[132] R. Subramoniam, D. Huisingh, and R. B. Chinnam, “Aftermarket remanufacturing strategic planning decision-making framework: Theory & practice,” *J. Clean. Prod.*, vol. 18, no. 16–17, pp. 1575–1586, 2010.

[133] M. Liu, C. Liu, L. Xing, F. Mei, and X. Zhang, “Study on a tolerance grading allocation method under uncertainty and quality oriented for remanufactured parts,” *Int. J. Adv. Manuf. Technol.*, vol. 87, no. 5–8, pp. 1265–1272, 2016.

[134] M. I. Mazhar, S. Kara, and H. Kaebernick, “Remaining life estimation of used components in consumer products: Life cycle data analysis by Weibull and artificial neural networks,” *J. Oper. Manag.*, vol. 25, no. 6, pp. 1184–1193, 2007.

[135] C. Okoh, R. Roy, J. Mehnen, and L. Redding, “Overview of Remaining Useful Life prediction techniques in Through-life Engineering Services,” *Procedia CIRP*, vol. 16, pp. 158–163, 2014.

[136] Y. Hua, S. Liu, and H. Zhang, “Remanufacturing decision based on RUL assessment,” *Procedia CIRP*, vol. 29, pp. 764–768, 2015.

[137] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1st ed. Cambridge: MIT Press, 1998.

[138] E. Sundin and B. Bras, “Making functional sales environmentally and economically beneficial through product remanufacturing,” *J. Clean. Prod.*, vol. 13, no. 9, pp. 913–925, Jul. 2005.

[139] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*, 2nd ed. New York: Cambridge University Press, 2017.

[140] T. Li, Z. C. Liu, H. C. Zhang, and Q. H. Jiang, “Environmental emissions and energy consumptions assessment of a diesel engine from the life cycle perspective,” *J. Clean. Prod.*, vol. 53, pp. 7–12, 2013.

[141] Y. Wang, A. Huang, C. A. Quigley, L. Li, and J. W. Sutherland, “Tolerance allocation: Balancing quality, cost, and waste through production rate optimization,” *J. Clean. Prod.*, vol. 285, no. xxxx, p. 124837, Feb. 2021.

[142] Y. Wang, L. Li, N. W. Hartman, and J. W. Sutherland, “Allocation of assembly tolerances to minimize costs,” *CIRP Ann.*, vol. 68, no. 1, pp. 13–16, 2019.

[143] Y. Wang, M. Perry, D. Whitlock, and J. W. Sutherland, “Detecting anomalies in time series data from a manufacturing system using recurrent neural networks,” *J. Manuf. Syst.*, no. December, 2020.

[144] Y. Wang, G. Mendis, S. Peng, and J. Sutherland, “Component-Oriented Reassembly in Remanufacturing Systems: Managing Uncertainty and Satisfying Customer Needs,” *J. Manuf. Sci. Eng.*, vol. 141, no. 2, pp. 0210051–02100512, 2019.