# PRODUCTION STARTING TIME CONTROL FOR COMPENSATING FORECAST ERROR AND CUSTOMER LOSS IN WAITING

**QIAN CHEN**

**Ph.D**

**The Hong Kong Polytechnic University**

**2014**

THE HONG KONG POLYTECHNIC UNIVERSITY
DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

# PRODUCTION STARTING TIME CONTROL FOR COMPENSATING FORECAST ERROR AND CUSTOMER LOSS IN WAITING

**QIAN Chen**

A thesis submitted in partial fulfillment of the requirements for Degree of Doctor of Philosophy

August 2013

# CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________ (Signed)

QIAN Chen (Name of student)

Abstract of thesis entitled

'Production Starting Time Control for Compensating Forecast Error and Customer Loss in Waiting'

Submitted by QIAN Chen

For the degree of Doctor of Philosophy

At The Hong Kong Polytechnic University in 2013

---

## ABSTRACT

Production systems play a key role in modern society, and significant improvements have been achieved through the years. However, due to the diversity of human behavior, customer demand uncertainty does exist in practice. Consequently, production solutions that are capable of coping with such diverse behavior are necessary. To have a better competitive position, forecasting is an important element in production management. Thus, there is always research for improving forecasting accuracy and the development of new methods is on-going.

Apart from working on new forecast methods or improving existing models, this research focused on working with the expected forecast error in a most economical way. To achieve this goal, the aim was to achieve a balance between the effect of forecast error with time and the customer loss in the waiting period. In this research, a production approach named Make-to-Balance (MTB) is introduced.

To verify the concept and the operating result of the proposed model, a simulation process was built with STELLA, and a software program was also coded in C# language. The SETLLA results and program results match well in different situations (one general case and four extreme cases) and identify the correctness of the MTB model. The program eases the calculations and it was found in this research

that the optimal solution could be obtained from MTB and Smart-MTB version programs. Indeed, the contributions of this research are not only in its inspirations but also in that it extends the view on how to run a production system effectively by taking uncertainty and customer behavior into account, and it also shows that customer loyalty helps to reduce the effect on forecast error.

## PUBLICATIONS ARISING FROM THE THESIS

**Journals**

1. Qian, C., Chan, C.Y. (2014). A new production approach for compensating forecast error and customer loss in waiting, *International Journal of Production Research*, DOI: 10.1080/00207543.2014.918292
2. Qian, C., Chan, C.Y. and Yung, K.L., (2011). Reaching a destination earlier by starting later: Revisited, *Transportation Research Part E*, 47, 641 – 647.
3. Qian, C., Chan, C.Y., Tang, C.S. and Yung, K.L., (2011). System monitoring through element flow reasoning, *Robotics and Computer-Integrated Manufacturing*, 27, 221–233.

**Conferences**

1. Qian C., Chan C.Y. and Gao J., (2011). Deployment of RFID in an Ocean Shipping Company, *International Journal of Arts & Sciences*, 4(12), 115–122.

## ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere thanks to my supervisor, Dr. C.Y. Chan, for his support, guidance and encouragement, which made this research project possible. Dr. Chan has been a source of perceptive and critical comments and moral support throughout the project. His enthusiasm and discipline gave me strength and encouragement. I would like to thank my co-supervisor, Prof. K.L. Yung for his suggestions and comments.

I would also like to thank my parents for the support they have provided me through my entire life and, in particular, I must acknowledge my wife; without her love and encouragement, I would not have finished this thesis.

The work described in this research project was supported substantially by a grant from the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University and the Research Grants Council of the Hong Kong Special Region.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# Chapter 1 Introduction

Manufacturing is vital to human society as it offers necessary products for a variety of mankind's activities. In the highly competitive situation today, it is necessary to understand how a production system should be designed before it is put into operation.

This research focused on the development of an effective production solution to address two issues: the forecast error and the customer loss in waiting. Forecasting is an important tool to reduce the customer loss due to waiting, but it is subject to error in such a way that the error becomes more serious as the projected timeline increases. It is essential to address the goal of how to strike a balance. The first chapter of this thesis gives a brief overview of production development history as well as some present challenges in production. Based on this background, the motivation for this research is spelled out and the objectives are defined.

## 1.1 Introduction to Production System Developments

Production not only increases the wealth of society, but also changes the nature of the world. In the late 18$^{th}$ century, the first industrial revolution started in Britain and dramatically changed the nature of production systems. With a series of innovations such as *the flying shuttle*, *the spinning jenny* and *the water frame*, and especially *the steam engine* that was developed by James Watt in 1765, production was freed from water power and location limits for the first time. Moreover, this made mass production in centralized locations possible and occurred in parallel with the development of the so-called fabrication system, where factory workers were organized based on new principles for the division of labor. As the consequence of the industrial revolution, production efficiency was improved significantly and

Britain transited from an economy based on agriculture to one based on industrial activities; it became 'the world's factory'.

It is also necessary to mention the second industrial revolution. The innovation in transportation (railroads) and communication (telegraphs) between 1850 and 1880 triggered the second industrial revolution. The introduction of standardized and interchangeable parts laid a good foundation for assembly system developments and also led to extensive mass production. At that time, America led the way in production innovations and, consequently, had more large-scale enterprises than all the other countries combined. Although high-volume production became common in many industries at the beginning of the 20$^{th}$ century, Henry Ford made the modern mass production possible with his great contribution to the concept of the moving assembly line. The most successful and well-known case was the Model T car, that started production in 1908. Through continual improvements and adopting the "moving assembly line" in Ford's factory, the labor time of a Model T car was reduced from 12.5 to 1.5 hours and the price also decreased to $290 by the 1920s. Subsequently, about 15 million Model T cars were sold and, in the early 1920s, the Ford Motor Company commanded two-thirds of the American automobile market.

With the growing market at that time, there were more demands on the mass production approach and General Motors (GM) took the lead. When DuPont and Sloan moved to GM, they re-identified the operation structure, including departments and motor products. With the newly developed, sophisticated procedures for demand forecasting, inventory tracking and market estimation, GM had taken nearly a 50 percent share of the USA motor market by 1940, far beyond the Ford Company.

The Toyota Production System (TPS) represented another paradigm shift during the second half of the 20th century. With the purpose of waste elimination in the early 1960s, Toyota launched the TPS that relied on the application of Kanban cards to prompt demand to the supply chain from the production point of view. The key idea of this is all about Just-in-time (JIT), inspired by the American supermarket system where products on the shelves bought by customers are soon replaced. Just-in-time, or the pulling principle, is concerned with producing products with minimal stocking. The system's distinctive practices have been addressed widely in the literature (Lander and Liker, 2007). TPS has achieved great success and has made Japanese manufacturers more efficient and reduced their production cost substantially (Hino, 2006). **Figure 1-1** shows some important milestones in production system developments.

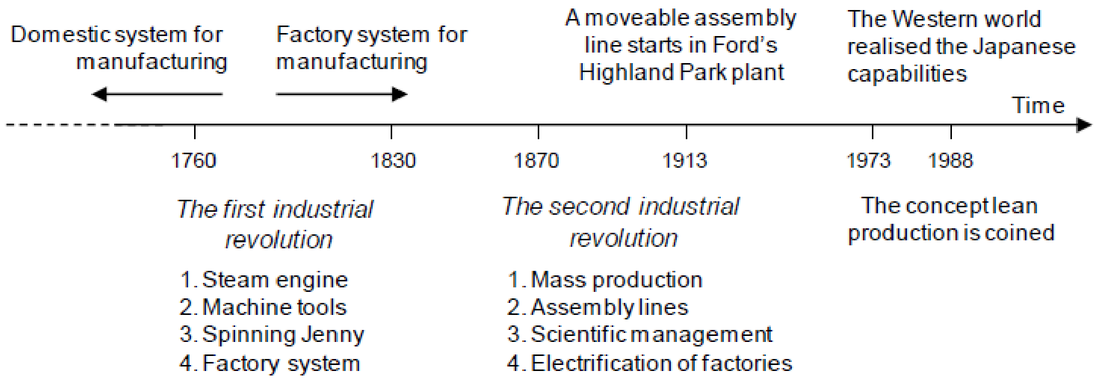



**Figure 1-1. Important milestones in the production system developments**

History shows that production systems are evolving from simple craft production to mechanized production, to mass production, etc., and such development is still continuing. The great leap in the industry was the mass production that enabled the fabrication of a limited variety of standardized products with a relatively low cost. A production system comprises a number of elements between which there are reciprocal relations. Commonly mentioned elements are premises, raw materials, people, machines or equipment and management. A structural perspective of the production system can be used to describe the different system elements and their relations, as in **Figure 1-2**. Each element is an important resource that is also a potential source of variation/disturbance which may seriously affect the production system performance.

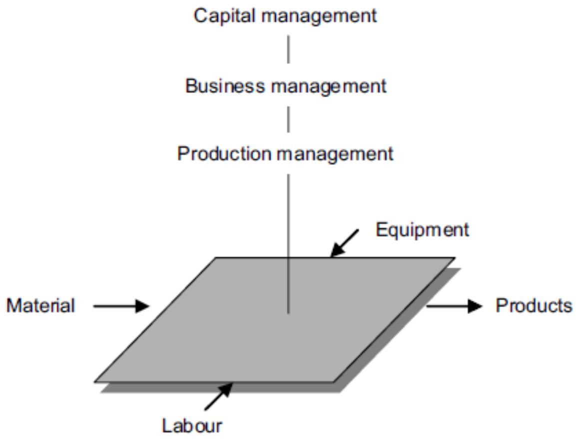



**Figure 1-2. Perspective production system model**

**Figure 1-3** briefly outlines the function blocks of a production system. Normally, the production schedule, which indicates what to do and in which time frame, is generated based on demand forecast or customer orders. With the production schedule, a production plan is created and then the manufacturing operations involved are carried out, following the established production plan strictly. Typically, the products are transferred to the inventory before they are shipped to customers. This cycle keeps going, with new schedules repeatedly prepared based on the market demand.

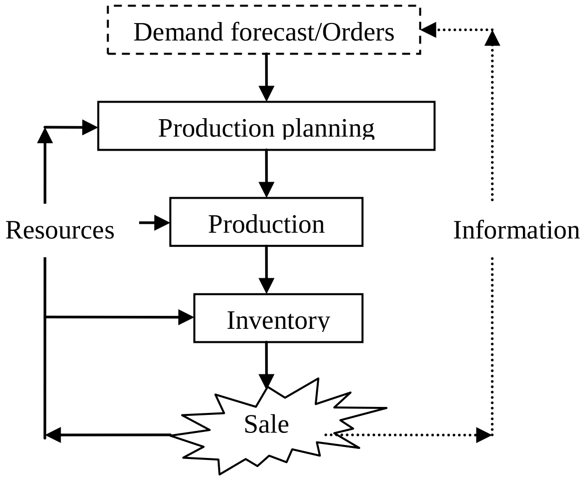



**Figure 1-3. Production system functions blocks**

## 1.2 Classical Production Control Approaches

There are some popular production approaches, such as the Make-to-Order (MTO) that is sometimes referred to as the Build-to-Order (BTO) (Gunasekaran & Ngai, 2009) or the Assemble-to-Order (ATO) (Tsai & Wang, 2009), and the Make-to-Stock (MTS) that may also be called the Build-to-Stock (BTS) (Wikipedia, 2013) and the hybrid MTO/MTS.

MTO is a production approach in which products start to be manufactured when a confirmed order has been received (Holweg *et al.*, 2001). It is a traditional production style for highly customized or low volume products. This approach is considered good for highly configured products like high-end bicycles, computer servers, or for products where holding an inventory is very expensive, such as aircraft (Parry & Graves, 2008). MTO has the advantage in a high product variety environment of supplying a product only when required by a customer, with low holding cost, but it usually has low manufacturing capacity utilization since the operation is initiated by a customer’s order.

In contrast, MTS is the dominant approach being adopted in many industries today and it refers to products manufactured before a final demand has been identified, with production volume driven by historical demand information (Parry & Graves, 2008). While providing a rapid response to customer demand, this approach is quite expensive, mainly in terms of stock/inventory, as well as transportation, because it is extremely tricky to determine the right quantity of products, where they are required at the right time. Moreover, it is often too fragile to cope with an unforeseeable situation.

With the purpose of gaining advantages for both the MTO and the MTS, the hybrid MTO/MTS came into being. This allows a manufacturer with low value/high volume parts to employ MTS, while a high value/low volume manufacturer will go for MTO. Since the main business objective of a company is to achieve long-term profitability, a manufacturer has to consistently produce high quality and specific products with competitive costs and high service levels. Thus, there is a driving force for moving from MTS to MTO and consequently, utilizing the hybrid MTS/MTO helps with the transition(Rabbani, 2010). Soman *et al*. (2004) provided the conceptual hierarchical production planning framework for a MTO/MTS production, as in **Figure 1-**. The most important strategic decision, also one challenge in MTS/MTO systems, is the MTS/MTO partitioning(Teimoury & Fathi, 2013). However, the microscopic view indicates that the transition from MTS to MTO may not be a simple switch over, because partial MTS can help to digest the forecast error and, subsequently, the transition can be made smoothly. In fact, finding the correct method to keep the stock/inventory to address the market demand and the operational stability properly is an academic research topic that has raised a lot of interest by researchers.

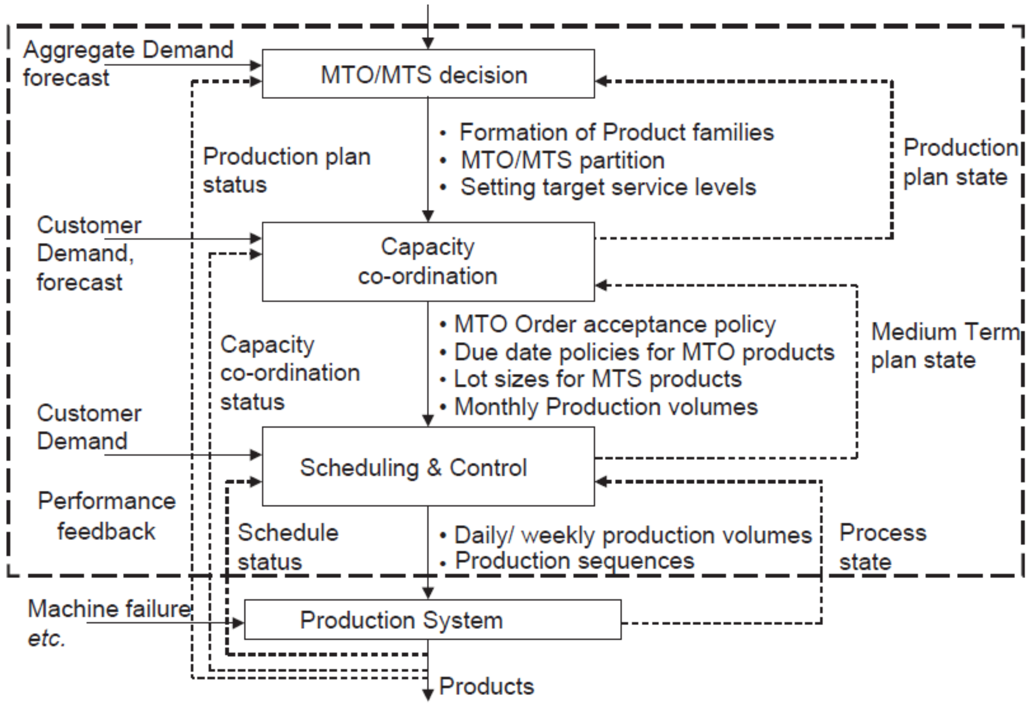



**Figure 1-4. Hierarchical MTO/MTS approach** (Soman *et al.*, 2004).

## 1.3 Challenges in Production

The advancements in IT have shortened the communication gap drastically and, hence, the market information can be obtained more quickly today. However, the diversity of customer demand is also increasing and this gives a challenge for production to enhance flexibility so as to simulate customers' requirements for a large variety of products; yet, this seems to be working against the benefits brought about by the large volume production. In fact, customers' minds keep changing and this surely creates stress to complete a production in time. Thus, having a fast response to customer needs is clearly always desirable. The instant reactions are to shorten the throughput time and to make the processes work faster, but as mentioned in the previous section, a higher production rate may end up with higher WIP, resulting in more material cost being locked up on the production floor. Furthermore, with the increasing uncertainty in customer demands, the probability of final product transfers to inventory is due to the failing of sale increases. Therefore, a good match between the response to customer demand and the production time taken is another challenge to tackle.

Chan *et al*. (2007) showed that the total production duration ($P_T$) for a batch with quantity ($Q$), throughput time ($T$) and production rate ($P_r$) was governed by the equation $P_T = T + (Q/P_r)$. Therefore, the throughput time and the production rate are major factors in determining how fast the production system can produce the required quantity to feed the market. Although an accurate customer demand forecast can help with some products being produced in advance, this needs production to be able to take in possible changes as fast as possible or at least under periodical reviews, even if it is considered not feasible to have an immediate response.

In short, increasing product variety will be a way to gain a better market share. Reducing the throughput time and fastening the production rate surely facilitates the reaction time to market changes, and results in the lowering of the production risk. Solely relying on forcing the production to have a higher production rate may not be useful if the throughput time is not taken into consideration, as this may end up by introducing more buffer stocks. Subject to the constraint of how to reduce the throughput time due to the production capability restriction, good forecasting can be a powerful tool to satisfy customer demand in a timely manner.

## 1.4 Observations from the Restaurant Production Approach

In real life, there is a common but specific production approach which is used in restaurants. A restaurant makes dishes to serve waiting customers who have ordered them; this production approach is equivalent to the MTO. Actually, the meal variety is not small, but having some well-prepared raw food materials cuts down the throughput time. Indeed, the preparation work is also part of the production, and this part of the production approach is similar to the MTS. As a result, the WIP increases, but these semi-finished goods in the kitchen facilitate the final operations significantly.

Another interesting observation is that there seems to be no special scheduling technique involved in the whole food production progression, since the system reacts directly to a customer order. Actually, most restaurants work on a first-come-first-served basis; that is some form of macroscopic production schedule, but is not the microcosmic schedule that we usually have in a typical production system. The schedule does not have detailed information at the beginning, such as when and what products (dishes or foods) should be made, only a general idea about the volume of different products that will be needed in a time period, and is just a good guess based on some historical knowledge. This makes a restaurant work well in terms of flexibility and it is also able to react to the demand quickly. The merit of the restaurant production approach is its fast response to diversified customers' demands at very lower product inventory levels. The main point is to have a quick reaction to the demand that is the time from knowing what a customer wants to serving the dishes. However, this can only be part of the total throughput time since most preparation work has been done in advance, and the remaining part is the production time needed ($Q/P_r$) (Chan *et al.*, 2007). In order to achieve this, the WIP may need to be increased to accommodate possible changes, as some information is not totally reliable because we do not know exactly what will happen at a later time.

## 1.5 Research Motivation

Having a fast response to customers in a demand uncertainty situation will greatly enhance the enterprise with respect to its competition. It is valuable to absorb some ideas from the restaurant production approach to enhance industrial production systems. The restaurant production system case shows that it is possible to have flexibility in producing different products in a short time. The restaurant production has one important characteristic, the actual production start ahead of the actual customer demand being identified, and the reaction time to the demand is very short, normally counted in minutes. It is understood that very few customers are willing to wait a long time for a meal! In fact, the process will be suspended at the semi-final product stage until a customer confirms the dish order, and then the process continues to the final product. Therefore, the WIP will be high during the process period. Furthermore, if there are only a few customers or customer orders are less than catered for in the preparation, the inventory increases; on the other hand, customers will turn to other restaurants when there is a short supply. **Figure 1-5** shows the restaurant production stages.

It can be seen that if the time *waiting for customer* shortens to zero, then the whole production process is not suspended and the WIP decreases as well due to a shorter throughput time, as there will not be a second stage. This arrangement favors almost all industrial production. Another important inspiration is the *customer waiting* time in the last stage, as shortening the *customer waiting* time for final products is essential to keep the customers in the pool but this leads to more work having to be done in advance. Once the entire second stage has been eliminated, the whole production period cannot be reduced further if there is no change in the existing production technique.

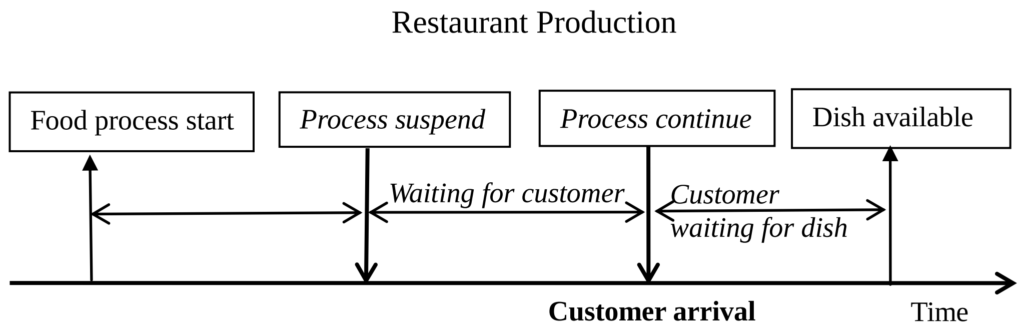



**Figure 1-5. Restaurant production stages**

## 1.6 Research Objective

The objective of this research is to establish a cost effective production control approach that is able to determine the most suitable production starting time by having a good balance between the forecast error and the customer loss due to waiting. Factors involved include the forecast production volume and its possible deviations, the production start ahead of time (before customers have arrived), the customer loss in waiting, the unit overproduction and underproduction costs, and the total production time required. Moreover, it was assumed that the production rate is unchanged in fabricating a product type in this research.

## 1.7 Research Achievement and Contribution

At the outset of this research, it was anticipated that the proposed production approach would be capable of having a better chance of minimizing the production venture by managing the production starting time more effectively. The outcomes will have a significant impact on the production control aspect and, more importantly, it may not be just as applicable for a particular production system. In fact, a short waiting time is what every customer wants and how a manufacturer can get it done in a cost-effective way is certainly a key to success. It is expected that the new concept brought out will benefit the operations of industrial systems.

## 1.8 Summary

Generally, there are two typical production control approaches, and they are: MTO or MTS. Then people also found that hybrid MTO/MTS can be more beneficial sometimes, and some works were done on finding the decoupling point. In fact, Assemble-to-Order (ATO) and BTO are also considered belonging to MTO, therefore, in this research MTO covers the both ATO and BTO. Today, customers want more personalized products leading to the increasing product diversity; MTO suits for this customer oriented situation, but customers may need to wait for a longer time. Moreover, vicious market competition is pushing manufacturers to respond to customers' needs quickly, and this means less waiting time is certainly an advantage in the marketplace. Obviously, MTS is good for this circumstance, but it has a higher chance of suffering loss caused by forecast error. Consequently, having a good compensation between the customer loss in waiting and the forecast error is important. The objective of this research is to determine the most suitable production starting time that may be somewhere between the MTO and MTS time horizon.

## 1.9 Thesis Outline

Apart from this chapter, there are five more chapters in this thesis. As a continuation of the research background, the methods of dealing with demand uncertainty in production planning, related control policies and key factors affecting production control (demand forecast and customer behaviour) are reviewed in Chapter 2. Chapter 3 introduces the new MTB concept and the main differences between it and the MTO/MTS. Formal definitions of the MTB framework, the associated mathematical modelling and how to determine the optimal production start ahead time are also presented in Chapter 3. A simulation model was developed by using the commercial software STELLA, mainly to serve the aim of examining the model's accuracy and the software coding for calculating the optimal production start ahead time. This was based on the proposed MTB model done by a program named *Smart MTB*, while another program called *MTB* that uses a step-by-step iteration approach to plot the whole curve; these are all introduced in Chapter 4. The verification and testing of the proposed MTB model are described in Chapter 5, where an illustrative example is also provided. Finally, discussion and concluding remarks are given in Chapter 6.

# Chapter 2 Literature Review

Research in production and operation management in relation to production control is reviewed in this chapter. Details of the control approaches: MTO, MTS and the hybrid MTO/MTS, are considered in Section 2.1. Section 2.2 reviews the two important factors affecting the production control; forecasting and customer behaviour. In addition, the some approaches to improving customer loyalty or patience are discussed.

## 2.1 Demand Uncertainty in Production Planning

One of the managerial challenges is that modern manufacturing is facing an uncertain demand caused by the bullwhip effect through the global supply chain (Lee *et al*., 1997). In reality, the demand uncertainty can be reduced effectively through appropriate demand aggregation and forecasting (Aigner *et al*., 1974). Many efforts have contributed to improvements to the demand forecast, such as MA(1), AR(1) and VAR(1) (Chen *et al*., 2010). In addition, there have also been many other researchers working on the generation of competent production plans. Meybodi *et al*. (1995) presented a hierarchical model to facilitate production planning and scheduling under a situation of random demand and possible production failure. Linear programming was proposed to minimize the total cost and workforce fluctuations, and to maximize the customer service level.

In terms of mastering the demand uncertainty, Gfrerer *et al*. (1995) suggested two production arrangements, forecast-driven production and customer-order-driven production. They also proposed a two-level (aggregation and disaggregation process) hierarchical model with a mathematical procedure for the forecast-driven production. Zapfel (1996) further enhanced the model by incorporating it into the MRP II, and later replaced forecast-driven production by customer-order-driven production to eliminate the demand uncertainty (Zapfel, 1998). However, to ensure customer-order-driven production works well in the case of uncertain demand, adequate production capacities must be made available so that every possible demand profile could have a feasible solution. This restriction seriously holds back the implementation of customer-order-driven production, especially in the long lead-time end-product industries. Furthermore, this method can neither shorten the customer waiting time nor improve the service level. To manage the raw materials shortage under demand uncertainty, Bertrand *et al*. (1999) investigated three planning procedures called the optimal, the deterministic and the myopic procedures. They argued that, in many practical situations, the best solution was to optimize the production with regard to the accepted customer orders. Spitter *et al*. (2005) studied the production timing within the planned lead-time and found that producing early had the benefit of lower safety stocks and higher utilization rate of production capacity. Another benefit was the ease of handling a demand fluctuation, but there was a disadvantage of high WIP inventory cost. If the demand was free of forecast error, Clark (2005) found that the scheduling horizon should be extended to 3 to 5 periods ahead of the production time, because the production would have a better performance with a longer planning horizon, but the cost tended to rise in line with the demand forecast error.

To address the demand uncertainty, a dynamic planning method was developed by Raa *et al.* (2005), with a subset of possible scenarios based on both the static and probabilistic approaches. By taking the inventory (or backlog) level into account, it updates the production quantity by subtracting the amount left in the inventory, or adding the backlogged amount. The comparative result showed that the dynamic planning solution had a smaller cost than those in statistic stochastic optimizations. Byrne *et al*. (2005) extended the linear programming approach by incorporating JIT concepts (the unit load concept and the effective loading ratio), combined with simulation to improve performance.

In the area of uncertainty in ERP, Grabot *et al*. (2005) discussed three ways to deal with the safety margins in quantity and delay evaluation, plan periodic refreshment and safety inventory. They proposed a fuzzy logic model to manage the situation. Their experiments showed that the fuzzy method was feasible in practice and could be integrated in a homogeneous way with the MRP. Mula *et al*. (2006) gave a comprehensive review of the existing literature on production planning under uncertainty and classified the planning models in four categories: conceptual, analytical, artificial intelligence and simulation. They concluded that the fuzzy set theory was an appropriate methodology which could give a great advancement in the production planning system. In fact, the approach of adopting the fuzzy approach in production planning and scheduling to cope with uncertainty can also be found in many other studies (Grabot *et al*., 2005; Schultmann *et al*., 2006; Mula *et al*., 2010; Guillaume *et al*., 2011; Mula *et al*., 2007; Liang, 2008; Lan *et al*., 2009; Aliev *et al*., 2007; Jamalnia *et al*., 2009; Peidro *et al*., 2009).

A rolling horizon planning approach with two-stage stochastic programming to take care of uncertainty was proposed by Tolio *et al*. (2007) and the research concluded that re-planning with timely information could result in reduced lead times, stock-outs and operational costs. However, more frequent re-planning also leads to more frequent rescheduling that causes increased inventory and ordering costs. Jonsson *et al*. (2006) also observed a trend for manufacturers to have more frequent re-planning. For example, weekly MRP re-planning was most common by

1993, but daily re-planning was more common by 2005. Hozak *et al*. (2009) conducted a literature review on re-planning frequency and summarized that there was a research-practice gap between much of the production planning research and the current empirical trend of having more frequent re-planning, and was more successful with increased rescheduling.

Two-stage and multi-stage policies have also been investigated. Wu *et al*. (2010) proposed a two-stage model to optimize price and demand uncertainty in the TFT-LCD industry. The essence of this approach is making production preparation (raw materials and other recourses) according to forecast results when the production is being carried out to minimize the impact of uncertainty. Zhang *et al*. (2011) presented a two-stage stochastic production planning model with the goal of minimizing the total production, inventory, and overtime costs under most scenarios. Zanjani *et al.* (2010) proposed a multi-stage stochastic program for production planning by separating the demand and yield uncertainties, and then integrated them with a stochastic programming model. The numerical results showed that their approach gave better results than the two-stage stochastic approach. Aghezzaf *et al*. (2010) discussed the tactical planning problem concerned with the determination of the optimal amount for each product-family at every manufacturing stage so that individual demand of the finished products was satisfied over the planning horizon with minimal inventory and cost. Three alternatives were recommended to generate production plans, namely the two-stage stochastic optimization model, the robust stochastic optimization model and the robust deterministic model. However, all are only concerned with the product quantity and the production capacity, but the fundamental question of monetary reward has not been considered!

## 2.2 Production Arrangement Approaches

A proper production arrangement is crucial in running a production plant with the intention of obtaining the best financial result. Thus, many production arrangement approaches have been developed such as the MTO, the MTS, and the hybrid MTO/MTS.

The well-known MTO and MTS are widely implemented in many industries, and there have been many research papers presented in this field (Yingdong, 2001; Dobson *et al*., 2002; Chang *et al*., 2003; Perona *et al*., 2009). A pure MTS production system is usually managed by demand forecast. Hence, products are processed in advance and stocked in a warehouse before a customer order arrives. For this system, a production line is designed for standard products and performance criteria are built upon a higher fill rate, demand forecasting, lot sizing, average work-in-process, etc. (Soman *et al*., 2004). In contrast with an MTS, MTO production is triggered only when a real order has been received from a customer. The criteria in an MTO system are average response time, average order delay, delivery lead-time, due dates, etc. (Soman *et al*., 2004). Haskose *et al.* (2004) discussed the arbitrary queuing network problem in an MTO manufacturing environment in which each workstation had limited buffer capacities. The research investigated the workload control in MTO and presented an approximation model with algorithms for getting the steady-state solution in any form of queuing network analytically.

To decide which approach has higher efficiency and better performance for a production system is a big challenge for manufacturers. Some adopt MTO to reduce the inventory-related costs, but others employ MTS to cut production runs and ensure a quick response time to customers. Hallgren *et al*. (2006) pointed out that the distinction between MTS and MTO operations is all about the selection of the manufacturing focus.

Rajagopalan (2002) presented a solution model for deciding whether a set of items should be made to stock or made to order, and the production policy for the MTS items in the environment is characterized by a large number of items with stochastic demand and varying processing times, significant changeover times and limited production capacity. The heuristic solution approach provides near-optimal solutions. Kumar *et al*. (2007) did a case study on a contract manufacturer regarding an analysis of MTO versus MTS. The result was that a make-to-stock policy is better than a make-to-order policy in terms of operating income for a single SKU (product).

Shao *et al*. (2012) evaluated MTO and MTS with order-fulfillment performance measurement in a system with an inventory cost budget constraint. Their research results showed that MTO was preferred in a production system with low component values and long component processing times or high values and short lead times, while MTS was applicable to a production system with either high component values and short component processing times or small values and long lead times. Nevertheless, this categorization method requires production to be distinguishable, but qualitative types of operation do not seem to be quite straightforward in yielding a proper classification result.

To integrate the advantages of both MTS and MTO, the hybrid MTS/MTO has attracted the attention of researchers in recent years. Chang *et al*. (2003) developed a heuristic production activity control (PAC) model to achieve a rigid order release plan and dispatching control in the hybrid MTO/MTS in a wafer production environment. Soman *et al*. (2004) presented a three-level general framework to manage such a system in the food processing industry. Later, Soman *et al*. (2006) used four different scheduling methods in MTS and hybrid MTO/MTS under stochastic demand, and compared their performances with extensive simulation studies. The results showed that the one that performed well in pure MTS definitely did not also perform well in the hybrid MTO/MTS. In 2007, a real case about the implementation of production planning and scheduling framework for a firm in the Netherlands producing 230 product types on a single line with limited capacity was provided by Soman *et al*. (2007). In this case, a few possible analytical decision aids were described and a heuristic for the MTO/MTS short-term batch-scheduling problem was also reported.

How to design a hybrid MTS/MTO system and solve related problems have also been examined. Donk (2001) adopted the *decoupling point* concept from (Hoekstra *et al*., 1992) to classify products into either MTS or MTO, based on some specific characteristics. Furthermore, a method for designing an efficient production system for combining both MTS and MTO products was also presented by Tsubone *et al.* (2002), in which the buffer capacity was a design variable for determining the production capacity at a higher planning level, while the rules for allocating production capacity to products were another design variable at a lower level. Dobson *et al*. (2002) also developed several models for jointly optimizing product offering, pricing, MTO versus MTS and cycle time decisions. The research results showed that three factors, the product holding costs, the customers' sensitivity to prices and delivery times, would impact upon the MTS versus MTO decision under the simplification of an infinite production rate in representing the inventory holding costs.

In analyzing a multi-product production/inventory system where demand on each item arriving is according to a Poisson process and the production time for each product has an Erlang distribution, Ohta *et al*. (2007) showed the optimality condition for MTO and MTS policies, as well as the optimal base-stock level of the MTS policy using the M/Er/1 queuing model instead of the M/G/1. A dynamic production control and scheduling model was presented by Eivazy *et al*. (2009), which encompassed two major modules, the release and dispatching modules for a semiconductor shop with the hybrid MTS/MTO production policy. Their research also pointed out that the future research direction was to extend the hierarchical production planning (HPP) for hybrid MTS/MTO production systems. Rafiei *et al*. (2012) extended the research and addressed the tactical level of HPP in a hybrid MTS/MTO production system. A five-step model was proposed to make important decisions about the order acceptance/rejection policy, order due-date setting, lot-sizing of MTS products and determining the required capacity during the planning horizon.

To deal with the issues of product diversity, improving delivery reliability, customizing products, providing suitable production flexibility and reducing system costs, manufacturers often use different production arrangements according to how the customer orders influence their systems. Therefore, the concepts of Order Penetration Point (OPP) and Customer Order Decoupling Point (CODP) were introduced. OPP is a topic receiving increasing attention in the manufacturing strategy literature. It is the point in the production value chain at which a specific order is linked to a specific product. Therefore, this point divides the production activities into forecast-driven (downward OPP) and customer-order-driven activities (upward OPP)(Olhager, 2003). Olhager (2003) also outlined factors that affect the positioning of the OPP into three categories: market, product and manufacturing characteristics. Rafiei *et al*. (2011) addressed a hybrid MTS/MTO product delivery strategy with two important decisions: order partitioning and OPP locating. A fuzzy method was proposed to determine the location of the OPP for products with this hybrid strategy.

In terms of CODP, Hallgren *et al*. (2006) investigated the relationships between different approaches to manufacturing relative to the CODP. The dotted lines in **Figure 2-1** depict the forecast-driven production activities and the solid lines are customer-order-driven activities. Wikner *et al.* (2007) analyzed the supply chain dynamics downstream of the CODP and proposed an order book control logic in a mass customization MTO system, which could maintain delivery lead time targets with capacity limitation.

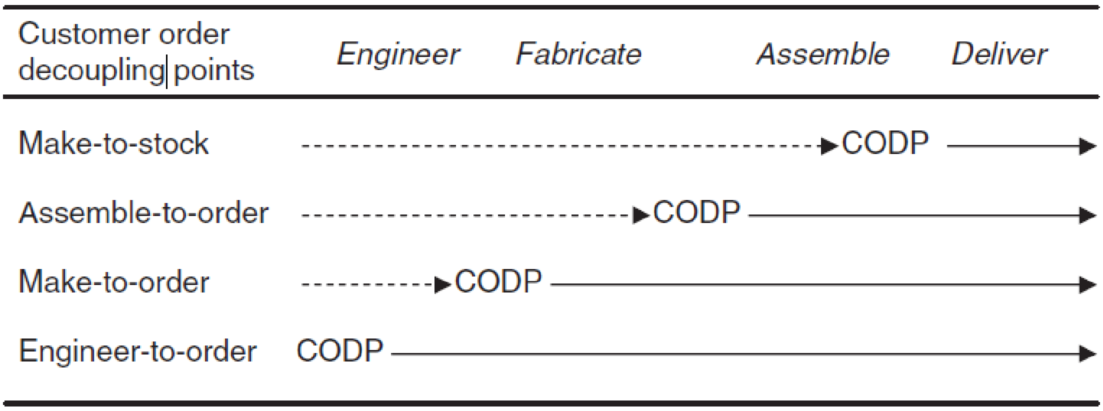



**Figure 2-1. Product delivery strategies related to CODP** (Hallgren *et al*., 2006)

Perona *et al*. (2009) developed a decision-making framework to plan inventory in small and medium sized enterprises (SMEs). The test results at a small steel wire producer showed that the new framework could provide a significant performance improvement when compared with a less formalized planning approach. However, the improvement was achieved in the trade-off between inventory cost, set-up cost and on-time delivery performance. Wu *et al*. (2008) extended the research work of Chang *et al.* (2003) on hybrid MTO/MTS, and presented a scheduling method for hybrid MTO/MTS semiconductor fabrication with imposed machine-dedication constraints to achieve a high on-time delivery rate for MTO products as well as a high throughput for MTS products.

Different characteristics have been analyzed in order to develop a framework for choosing the most suitable decoupling point and replenishment policy and for determining the parameters of the chosen policy, such as the classical Economic Order Quantity (EOQ). A multi-criteria decision-making method was proposed by Hemmati *et al*. (2010), in which relevant criteria affecting MTS/MTO partition were split into four categories, market-related criteria, product-related criteria, process-related criteria and supplier-related criteria. The prime aim was to choose appropriate product delivery strategies for different products in the manufacturing system. The results revealed that the most important factor was the "required delivery time" and the company preferred to utilize MTO policy for producing a new product. In addition, there was a hybrid MTS/MTO optimal production control policy developed by Iravani *et al*. (2012). With a partial-linear structure characterized by three parameters, the base-stock level, the rationing level and the admission level.

Indeed, there is a new hybrid MTS/MTO development direction that is to design a dynamic hybrid system in which the decoupling point in the MTS/MTO production capacity coordination can be re-adjusted according to the different demand patterns and customization levels. Zhang *et al*. (2013) suggested a multi-server queuing model of a dynamic hybrid system, where a subset of servers or machines is switched dynamically between MTS and MTO production via a congestion-based switching policy. The research results showed that the dynamic hybrid system generally outperformed the static hybrid system, particularly when the traffic intensity was high, in both MTS and MTO operations.

In fact, there has not been much research work on the hybrid MTO/MTS published in the literature, and the studies that have been published focused mainly on food processing and IT industries. Most were about how to decide the decoupling point of MTO and MTS or related problems, **Figure 2-2** is a schematic drawing of a hybrid MTS/MTO production line, where it can be observed that the difference in the decoupling point (the separation point of MTS and MTO) impacts not only on the planning/scheduling/control of production, but also the physical layout of a production line, in most situations.

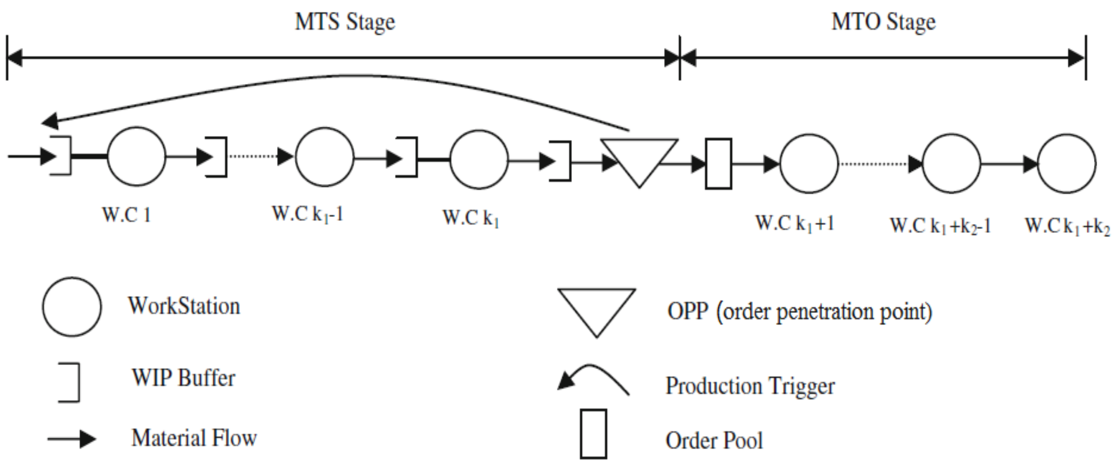



**Figure 2-2. Schematic of hybrid MTS/MTO production line**(Yingdong, 2001)

## 2.3 Key Factors Affecting Production Arrangement

There are many factors that affect the production arrangement, and forecast and customer behavior are two of definite importance. Although some investigations have been done on these two factors individually, the room still exists to study further details, such as their interaction. Thus, some related research work concerning forecasting and customer behavior is reviewed in the following sections.

### 2.3.1 Demand Forecast

Forecasting is a tool used widely in many fields, such as economics and finance (Elliott *et al*., 2008), tourism (Song *et al*., 2008; Goh *et al*., 2011), telecommunication (Fildes *et al.*, 2002), the medical industry (Wargon *et al*., 2009), construction(Wee, 2007), and market demand (Lee *et al*., 2006). It also plays an important role in modern manufacturing areas (Syntetos *et al*., 2009; Boylan *et al*., 2010). Manufacturers are very enthusiastic about having good forecasting support systems to have more accurate forecast results and, therefore, a great deal of effort has been made to improve the demand forecast models such as the MA(1), the AR(1), the VAR(1) and artificial neural networks (NNs) (Chen *et al*., 2010; Hippert *et al*., 2001). Alfares *et al*. (2002) examined a wide range of forecast models and classified them into nine categories: multiple regressions, exponential smoothing approaches, iterative reweighted least-squares, adaptive load forecasting methods, stochastic time series, ARMAX models based on genetic algorithms, fuzzy logics, neural networks and expert systems.

Other forecast-related issues were investigated as well, such as forecast evaluation (Granger *et al*., 2000) and forecast behavior (Leitner *et al*., 2011). Up to the present time, there have been enormous achievements in forecast methodologies and applications (Fildes *et al.*, 2008; Syntetos *et al.*, 2009; Song *et al*., 2008; Goh *et al*., 2011). Wee (2007) studied the quality of demand forecast in large-scale projects and suggested that the quality of the demand forecast was associated with the application of state-of-the-art techniques, the adoption of external views and the introduction of measuring tools. Recent studies also indicated that some advanced techniques and their combinations tended to result in better forecast accuracy under certain circumstances (Goh *et al*., 2011). However, there is no singular method that can outperform others, with clear-cut evidence, in forecast competition (Song *et al*., 2008). Some broad conclusions regarding the practical construction of a forecast case have been given: the forecast objective is important, time series models often change with time, a combination of forecasting methods may offer an attractive alternative and related guidance is important to give a good result (Elliott & Timmermann, 2008). Forecast applications in marketing operations had been studied by Fildes *et al*. (2008) over a period of 25 years. They pointed out that intermittent demand, sales responsiveness and computer-intensive methods applied to direct marketing and credit risk appraisal have proven susceptible to new ideas. In addition, knowledge of the bullwhip effect and the benefits of information sharing have provided a valuable opportunity for academic modelers.

Although substantial achievements in forecasting have been achieved, some forms of forecast error are still inevitable. A lot of the researchers have concentrated on how to improve forecast accuracy, but little attention has been given to how to relate the impact caused by forecast errors. Bert (2007) conducted a review of the quality of demand forecasts and spelled out that only a few studies had made a systematic examination between the forecast and the actual demand. Some factors that influenced the forecasting quality were investigated and it was shown that strategic behavior seemed to be more important than other factors in the demand forecast. Unfortunately, the research was on large projects only, without considering the “time” factor in association with the forecast result. Fildes *et al*. (2011) carried out a comprehensive study of the problem of demand uncertainty and forecast error in relation to the unit costs and the customer service levels in the supply chain field. Although they proposed a framework for examining the effect on forecast error, how to minimize the effect due to forecast error was not mentioned. In practice, if we accept that forecast error is unavoidable, then it will be sensible to address the issue of how to cope with this error.

### 2.3.2 Customer Behavior

Customer behavior has attracted a lot of attention for the reason that it affects the sales directly and this is fundamentally essential for all companies. Many models for investigating customer behavior have been formulated mathematically. The research can be divided into two main sectors. One is about keeping customers' satisfaction (or loyalty) and the other is to investigate the customer behavior.

Disconfirmation theory is concerned with the cognition of transaction specific experiences that is an underpinning for customer (dis-)satisfaction and subsequent behavior, while cognitive psychology is concerned with the importance of cognitive schemas in the decision process. In the service industries, disconfirmation of expectations and cognitive psychology are employed in predicting consumer behavior (Andreassen *et al*., 1998). A field study in a supermarket to find out the relationship between customer waiting time and satisfaction with the server was conducted by Gail *et al*. (1997). The findings were that satisfaction with the server increased with decrease in “perceived” waiting time, and socializing with customers helped to reduce the “perceived” waiting time. In the study of waiting time, Fitzsimons (2000) also found that some customers would wait for the item they originally sought while others would not, either because they decided to buy a substitute or they did not buy anything. Andreassen *et al*. (1998) examined the influence of company image and customer satisfaction on customer loyalty and concluded that, for complex and infrequently purchased services, corporate image rather than customer satisfaction was the main predictor of customer loyalty. This finding challenges the disconfirmation of expectations paradigm, which predicts customer satisfaction as the primary root of customer loyalty.

In the long tail phenomenon, Bardacki *et al*. (2004) and Chen *et al*. (2005) showed that customer behavior differs across markets. Brynjolfsson *et al*., (2011) and Elberse (2008) adopted the idea that customer behavior was heterogeneous in a market, and this was also supported by Elias (2002). Brabazona *et al*. (2012) assumed that there were two key factors affecting customer purchasing decisions, the amount of specification compromise that a customer was willing to make and the length of waiting time. This point was also evident from findings from other researchers (Elias, 2002; Holweg *et al*., 2004; Bardacki *et al*., 2004; Fredriksson *et al*., 2005). Besides waiting time and corporate image, many other factors were investigated, such as personal characteristics (Homburg *et al*., 2001), culture difference (Chen *et al*., 2005), organization's complaint management (Homburg *et al*., 2005), and service improvement (Kumar, 2005).

Typically, customer patience is analogous to the waiting time. Bae *et al*. (2001) studied the M/G/1 queue with impatient customers and proposed a formula for the limiting distribution of the virtual waiting time. Mandelbaum *et al*. (2004) made some observations in call centers regarding the relationship between customer patience with the M/M/n+G queue, and discovered that there was a linear relationship between the probability of abandoning and the average wait time prevailing, both practically and theoretically. Based on the study of extensive statistical data from a telephone call center, Brown *et al*. (2005) proposed a survival function of customers remaining on hold after having waited for a given length of time, and the research data showed clearly that it decreased monotonically from unity as the length of time increased. Anderson *et al*. (2006) analyzed the purchasing case of 20,000 customers of a mail-order catalog company and their findings revealed that the percentage of customers who cancelled their orders increased with the anticipated delay before the item was expected to be shipped. Gershwin *et al*. (2009) offered a defection function, similar to the function mentioned in Brown *et al*. (2005), and they indicated the fraction of potential customers who chose not to complete their orders when the backlog reached a given level. Veatch (2009) demonstrated that, for a simple model of production control, the impact of customer impatience on lost sales could be captured by a single parameter, a mean sojourn time, even when customer behavior was a function of the backlog. Robinson *et al*. (2011) studied the cost of a patient's waiting time in a doctor's office and gave an observation-based method for estimating the relative cost in association with the customer waiting time.

In a nutshell, there are many factors contributing to customer behavior, including waiting time, corporate image, personal characteristics, cultural difference, the organization’s complaint management, and service improvement. Moreover, in different markets (products, nations, countries, or even age groups), the effects of the aforementioned factors on customers are also quite different. Nevertheless, the amount of specification compromise and the length of waiting time are always two key factors affecting customer purchasing decisions. Furthermore, there is probably a linear relationship between the probability to abandon the purchase and the average waiting time.

## 2.4 Summary

This chapter reviewed four key areas related to the research: Demand uncertainty, Production control approaches (MTO/MTS/Hybrid), Demand forecast and Customer behavior. In terms of the demand uncertainty, it showed that there were generally three directions to deal with: improving demand forecast, adopting production planning technique and producing early.

In production approach research, the review indicated that MTO suited for low component values and long component processing times or high values with short lead times, while MTS was just the adverse case. However, it is hard to fit all production systems well into either MTO or MTS. Thus, the hybrid MTO/MTS comes to the interests of researchers.

The review in forecast hinted that no forecast method could be 100 percent accurate as there were too many elements involved and the forecast technique could be quite distinctive towards the case. Lastly, customer patience was analogous to the waiting time (satisfaction with the server increased with decrease in “perceived” waiting time). Typically, the amount of specification compromise and the length of waiting time were two key factors affecting customer purchasing decisions. Therefore, shortening customer waiting time is a purpose of production control.

# Chapter 3 Development of MTB Model

Manufacturers always have a strong desire to achieve the goal of providing more product varieties and having higher customer service levels in order to make the company more profitable. This chapter introduces a novel production approach named Make-to-Balance (MTB), which was inspired by the restaurant production control approach discussed in Section 1.3. With the purpose of having both the advantages of MTO and MTS so as to achieve better production performance without the complication of determining the decoupling point and the capacity coordination in the hybrid MTO/MTS, MTB provides a quick way to determine an optimal production arrangement by making a balance between the forecast error and customer loss in waiting.

## 3.1 Introduction to MTB Methodology Development

There are many factors affecting production operations, and MTB balances the influences of the two key factors to achieve an optimal result. In the following sections, the schematic framework and concept of this research are presented first, and the associated mathematical model is then described in detail.

### 3.1.1 Conceptual MTB Framework

Demand uncertainty may have a serious negative impact on production and therefore a great deal of effort has been put into this area. In fact, both MTO and MTS have the same purpose of minimizing the negative effect through different strategies, but previous methods have focused mainly on how to switch between MTO and MTS. However, there are other factors affecting production and, possibly, some interactions exist among these factors as well.

The fundamental importance of the new MTB production model is to achieve the optimal profit. **Figure 3-1** shows the conceptual framework. Although there are N factors involved and the dash lines mean that there are possibly some interactions among these factors, the proposed MTB production arrangement concentrates on resolving the conflict between two key factors, the forecast error and the customer loss in waiting. It is not difficult to see that the customers' patience is fading with time and forecasting is an essential tool to shorten the customers' waiting time, but it is always subject to unavoidable forecast error.

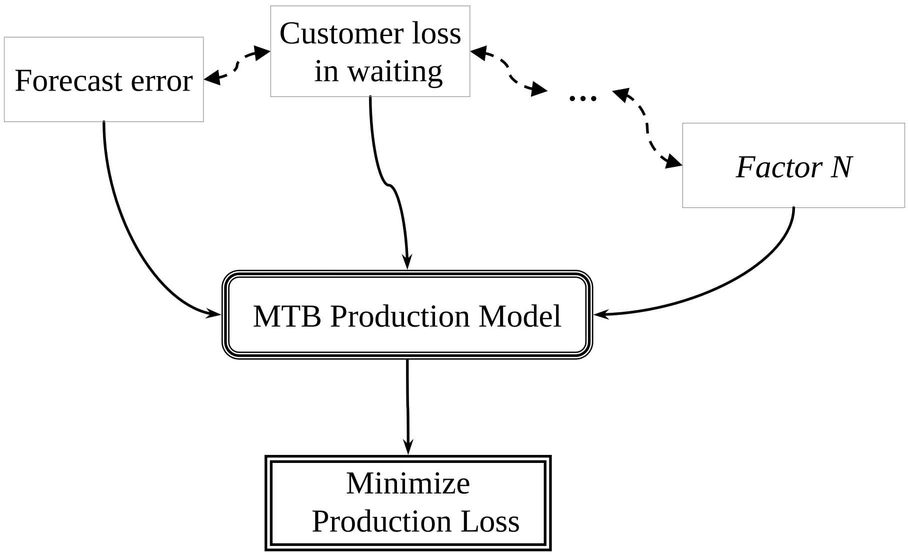


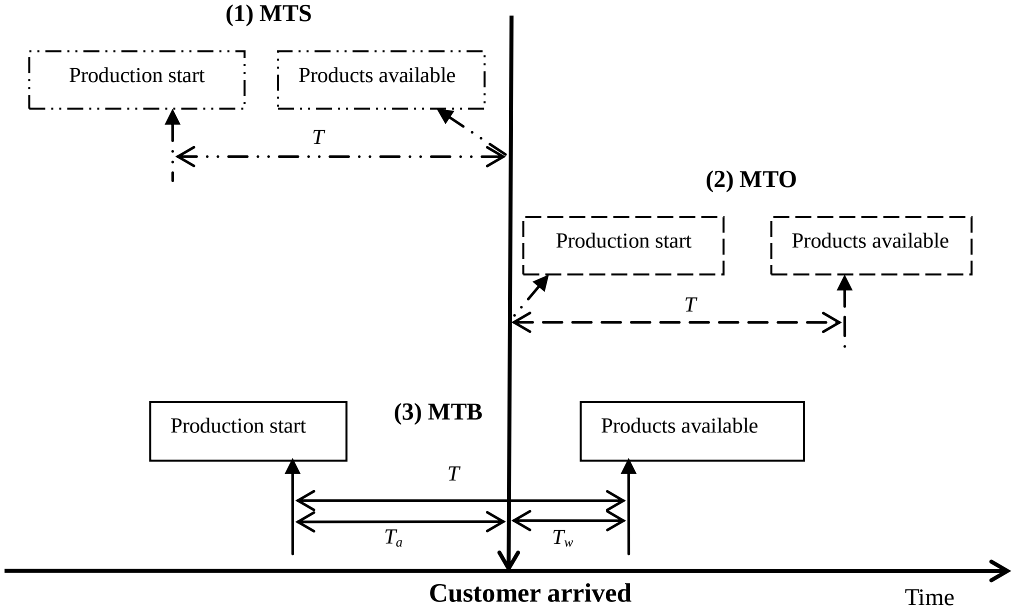

**Figure 3-1. Production optimization framework**

### 3.1.2 MTO/MTS/MTB Time Horizons

To improve the customer service level, it is important to produce the right product quantity at the right time, and therefore demand quantity and product availability time are the key elements. Forecasting and customer reaction are related closely and hence these two elements are involved in developing the MTB model.

The longer the time spans in a forecast, the larger the forecast error. With a longer waiting time, customers will be less patient and there is a higher possibility for them to change their minds. It can be observed that time is the common measure of forecast error and customer loss in waiting. Thus, MTB aims at identifying the most suitable time in the specific time window for the production to start at to have a good balance in customer waiting time and forecast error as well. **Figure 3-2** shows the time horizons of different production arrangements. They are labeled as: (1) the MTS production, (2) the MTO production; (3) the proposed MTB production.

In the case of the MTS arrangement, the production starts right ahead with the whole total time ($T$) or more (i.e., $T_a \geq T$, $T_w=0$) before the customer's arrival, and the product quantity is usually made by the forecast. The advantage of this production type is that the customer waiting time is the least. However, forecast error is generally difficult to prevent and hence it may lead to either an excessive inventory or a stock-out condition. In contrast, MTO only begins after the customer order has been confirmed and, therefore, excessive inventory or stock-out can usually be avoided. Yet, this arrangement implies that customers have enough patience to wait for the products. In fact, no customer has indefinite patience (Benjaafar *et al*., 2010) and generally, the longer the waiting time, the higher the chance of losing a sale. In fact, inventory problems and customer loss are situations that all companies want to stay away from, and so there is a dilemma regarding the forecasting period. This is because a shorter forecast time ($T_a$) increases the accuracy, but it also increases the customer waiting time ($T_w$) that leads to more likely ending up with more sale failures due to customer loss. Therefore, it is sensible to make a balance between the forecast duration and the customer waiting time in order to minimize the loss. Indeed, the development of the MTB production model is to accommodate such a case. **Table 3-1** gives an overview of the characteristics of the three production arrangements that have just been discussed.




$T_a$ - production start ahead time (before customer arrived);
$T_w$ - customer waiting time;
$T$ - total time (throughput time and delivery time) that also equals to ($T_a$ + $T_w$).

**Figure 3-2. Time horizons of different production arrangements**

**Table 3-1. Characteristics of production arrangements**

| Production arrangement | Production ahead time | Customer waiting time | Possibility of excessive inventory or stock-out | Customer loss |
|---|---|---|---|---|
| **MTS** | $T$ | $0$ | High | Low |
| **MTO** | $0$ | $T$ | Low | High |
| **MTB** | $T_a$ | $T_w$ | Moderate | Moderate |

Remark: $T = T_a + T_w$

## 3.2 MTB Mathematical Modelling

A supreme production system aims to produce the right quantity that meets the market demand at the right time. However, there are always uncertainties and they may bring some forms of loss to the company. The key purpose of the MTB model is to make the right compensation for the negative effects due to the forecast error and the customer loss in waiting with the specific time window ($T$) so as to minimize the loss. To formulate this model, it is necessary to establish the relationship between the forecast error and the customer loss in waiting; the basic mathematical notations involved in the model development are given in **Table 3-2**.

**Table 3-2. Mathematical notations**

| | |
|---|---|
| $T_a$ | Production start ahead time (before the customer arrives) |
| $T_w$ | Customer waiting time |
| $T$ | Total time required ($T = T_a + T_w$) |
| $X$ | Real demand quantity of products |
| $X_f$ | Forecast demand quantity ( = Produced product quantity) |
| $L_o$ | Loss of overproduction per product in cost |
| $L_u$ | Loss of underproduction per product in cost |
| $P_f(T_a)$ | Forecast error percentage function at ahead time ($T_a$) [$0 \leq P_f(T_a) \leq 1$] |
| $P_c(T_w)$ | Customer loss percentage function after waiting for time ($T_w$) [$0 \leq P_c(T_w) \leq 1$] |

Normally, there are two opposite trends expected from a forecast value. The first is the forecast demand (it is also considered as the quantity to be produced) that is greater than the real demand ($X$) with the production start ahead time ($T_a$) such that:

$$X_f = X + X \cdot P_f(T_a)^{+} \qquad [1]$$

Arguably, it could be more appropriate to use the following equation instead:

$$X = X_f + X_f \cdot P_f(T_a)^{+} \qquad [2]$$

However, this creates some difficulties in the model development and calculations have shown that the differences in these two error percentage functions can be determined as follows:

$$X_f = X + X \cdot P_{f1}\,(T_a)^+ \quad \& \quad X = X_f + X_f \cdot P_{f2}(T_a)^+$$

$$X = \frac{X_f}{1 + P_{f1}(T_a)^+} \quad \& \quad X = X_f\,(1 + P_{f2}(T_a)^+)$$

$$\frac{X_f}{1 + P_{f1}(T_a)^+} = X_f(1 + P_{f2}(T_a)^+)$$

$$1 = (1 + P_{f2}(T_a)^+)(1 + P_{f1}(T_a)^+)$$

$$0 = P_{f1}(T_a)^+ + P_{f2}(T_a)^+ + P_{f1}(T_a)^+ \cdot P_{f2}(T_a)^+$$

Assuming that the deviation in these two probability functions is $\Delta P$ and,

$$P_{f2}(T_a)^+ = P_{f1}(T_a)^+ + \Delta P$$

By substitution,

$$0 = P_{f1}(T_a)^+ + \left(P_{f1}(T_a)^+ + \Delta P\right) + P_{f1}(T_a)^+ \cdot \left(P_{f1}(T_a)^+ + \Delta P\right)$$

$$0 = 2P_{f1}(T_a)^+ + \Delta P + \left[P_{f1}(T_a)^+\right]^2 + P_{f1}(T_a)^+ \cdot \Delta P$$

$$\Delta P = \frac{-P_{f1}(T_a)^+\left[P_{f1}(T_a)^+ + 2\right]}{\left(P_{f1}(T_a)^+ + 1\right)}$$

Since $0 \leq P_{f1}(T_a) \leq 1$, the relationship of these two functions is plotted as in **Figure 3-3** and data is given in **Table 3-3**.

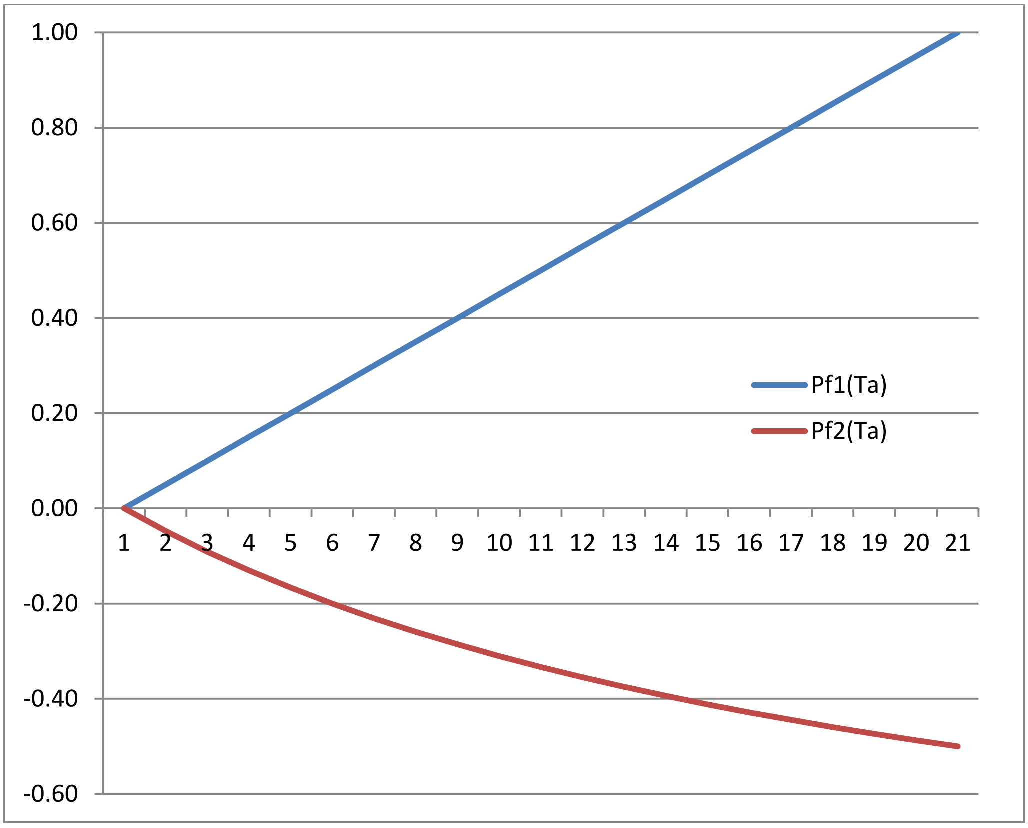



**Figure 3-3. Relationship between forecast functions**

**Table 3-3. Deviation of forecast functions**

| $P_{f1}(T_a)$ | $P_{f2}(T_a)$ | $\Delta P$ | | $P_{f1}(Ta)$ | $P_{f2}(T_a)$ | $\Delta P$ |
|---|---|---|---|---|---|---|
| 0.00 | 0.00 | 0.00 | | 0.55 | -0.35 | -0.90 |
| 0.05 | -0.05 | -0.10 | | 0.60 | -0.38 | -0.98 |
| 0.10 | -0.09 | -0.19 | | 0.65 | -0.39 | -1.04 |
| 0.15 | -0.13 | -0.28 | | 0.70 | -0.41 | -1.11 |
| 0.20 | -0.17 | -0.37 | | 0.75 | -0.43 | -1.18 |
| 0.25 | -0.20 | -0.45 | | 0.80 | -0.44 | -1.24 |
| 0.30 | -0.23 | -0.53 | | 0.85 | -0.46 | -1.31 |
| 0.35 | -0.26 | -0.61 | | 0.90 | -0.47 | -1.37 |
| 0.40 | -0.29 | -0.69 | | 0.95 | -0.49 | -1.44 |
| 0.45 | -0.31 | -0.76 | | 1.00 | -0.50 | -1.50 |
| 0.50 | -0.33 | -0.83 | | | | |

Similarly, in the case of $P_{f2}(T_a)^+ = P_{f1}(T_a)^+ - \Delta P$, we obtain:

$$\Delta P = \frac{P_{f1}(T_a)^+ \left[P_{f1}(T_a)^+ + 2\right]}{\left(P_{f1}(T_a)^+ + 1\right)}$$

It turns out that using $X_f = X + X \cdot P_f(T_a)^+$ is always able to cover the range of $X = X_f + X_f \cdot P_f(T_a)^+$ in terms of the error spread. Although they are on opposite sides, this is still applicable because the forecast error also spreads to both sides. Therefore, here will not be a negative effect by employing equation [2] to generate the "real demand" in simulation tests and so forth.

**Figure 3-4** is a graphical illustration of such a situation in which it takes a total time period of $T$ to produce the quantity of products forecasted. The solid line in the middle of the figure stands for the real customer demand while the solid line on the left side is the forecast demand, which is higher than the real one in this case. The horizontal displacement between the forecast demand and the real demand is $T_a$, that is the production starts at $T_a$ time ahead of the customer arrival. The solid line on the right side represents the customers remaining after the $T_w$ waiting period, where the products are available at that time. The difference in height between the right and left solid lines is the overproduction volume when the production is finished. Indeed, this is an overproduction situation with $(X \cdot P_f(T_a)^+)$ units overproduced. Moreover, there is another additional portion contributed by the customer loss of $(X \cdot P_c(T_w))$ units due to the waiting time $(T_w)$, and this also further stretches the loss in overproduction.

For value $C_1$ that represents the total loss in the situation where the forecast demand is larger than the real demand, plus the consequence of loss due to customer waiting, then:

$$C_1 = X \cdot P_f(T_a)^+ \cdot L_o + X \cdot P_c(T_w) \cdot L_o \qquad [3]$$

In the case of $c_1$ equals to $C_1$ per "product", it becomes:

$$c_1 = C_1/X = P_f(T_a)^+ \cdot L_o + P_c(T_w) \cdot L_o \qquad [4]$$

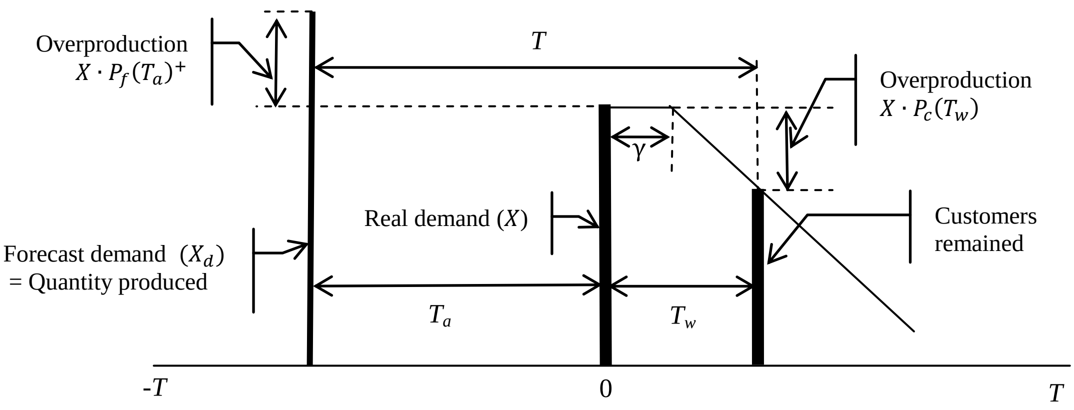



**Figure 3-4. Forecast demand larger than real demand**

The first part of $c_1$ is concerned with the loss of overproduction as a result of the forecast error. The second is the further overproduction contributed by the loss of customers in waiting.

In contrast, **Figure 3-5** illustrates the situation where the forecast demand is smaller than the real demand with the start ahead time $(T_a)$. Comparable to **Figure 3-4**, the solid line in the middle stands for the real customer demand whilst the solid line on the left side is the forecast demand but is lower than real demand. The rightmost solid line also represents customers remaining after $T_w$. Now, the difference between the left and the middle solid lines is the underproduction volume. The difference between the middle and left solid lines is the total loss in customers. The difference between the left and right solid lines is the overproduction volume caused, including the customer loss at the time when products are available.

In an underproduction situation with a shortage of $(X \cdot P_f(T_a)^-)$ units, the customer loss in waiting is analogous to the previous overproduction case, but when the customer loss level is smaller than the forecast error $(P_c(T_w) \leq P_f(T_a)^-)$, the underproduction situation prevails; but once $(P_c(T_w) > P_f(T_a)^-)$, the customer loss brings in an overproduction situation. Therefore, for the total loss $C_2$ when the forecast demand is smaller than the real demand plus the result of customer waiting loss, this turns out to be:

$$C_2 = X \cdot P_f(T_a)^- \cdot L_u + X \cdot Max\left[P_c(T_w) - P_f(T_a)^-, 0\right] \cdot L_o \qquad [5]$$

Similarly, $c_2$ equals to $C_2$ per product giving:

$$c_2 = C_2/\mathrm{X} = P_f(T_a)^- \cdot L_u + Max\left[P_c(T_w) - P_f(T_a)^-, 0\right] \cdot L_o \qquad [6]$$

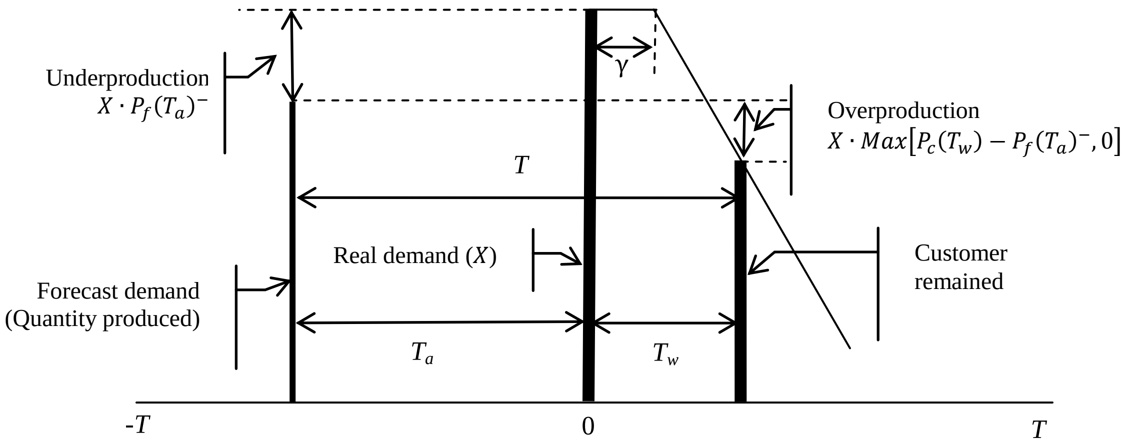



**Figure 3-5. Forecast demand smaller than real demand**

The first part of $c_2$ is caused by the forecast error and the second part is the possible overproduced quantity caused by customers leaving. If the possibility of overproduction is $\rho$, the objective function is:

$$Min[Z(T_a)] = \rho c_1 + (1-\rho)c_2 \qquad [7]$$

where $T_a$ is the most important factor in having a minimum Z value. Moreover, there are two more functions that need to be plugged into the current model: the forecast error percentage against time horizon ($P_f(T_a)$) and the customer loss percentage against waiting time ($P_c(T_w)$) functions. The following sections introduce the formulation of these two functions.

### 3.2.1 Forecast Error Spread Percentage Function Modelling

Normally, the forecast error spread decreases as the forecast time duration becomes shorter. Therefore, to have a better forecast accuracy, there is a need to reduce the production ahead time, but consequently, customers are required to wait for a longer time to receive the products/services. Although forecast methods were reviewed in Section 2.2.1, research on the mathematical modelling of the forecast error spread is still limited. Since point forecasts are often less appropriate than predictive distributions (Granger *et al*., 2000), the exponential function is adopted for modelling the error spread against time, and assumes that the chance is evenly distributed across the spread in a random manner. In modelling the forecast error spread with time, $\alpha$ represents the error parameter ($0 \leq \alpha \leq 1$) such that a larger value of $\alpha$ leads to a higher forecast error spread (see **Figure 3-6**). Thus, the forecast error range is +/-[0, $1 - e^{-(\alpha \cdot T_a)}$] at time $T_a$. Therefore,

$$P_f(T_a) = 1 - e^{-(\alpha \cdot T_a)} \qquad [8]$$

The forecast error can be positive, negative, oscillating from period to period, etc. As a result, the function $P_f(T_a)$ can be different in dissimilar forecasting mechanisms in order to accommodate the needs in various situations.

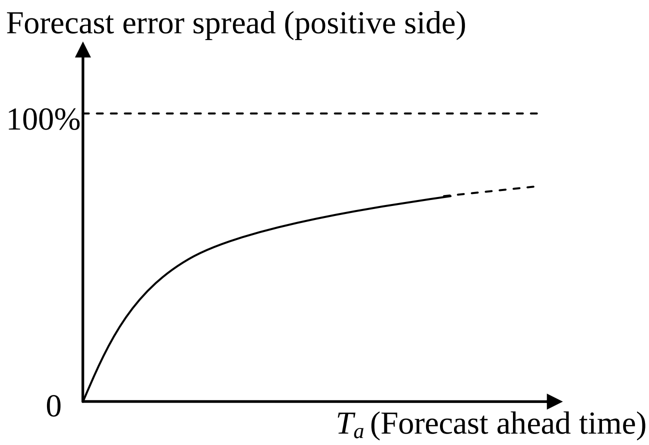



**Figure 3-6. Forecast error spread percentage against ahead time graph**

### 3.2.2 Modeling of Customer Loss in Waiting Percentage Function

Normally, a long waiting time bears more on a customer's patience, and there is a higher chance for a customer to change his/her mind (Taylor, 1994). To model this scenario, a linear relation between the probability to abandon against the waiting time can be used (Mandelbaum *et al*., 2004). In **Figure 3-7**, the customer loss percentage is a piecewise linear function modeled as:

$$P_c(T_w) = \begin{cases} 0 & (T_w \leq \gamma) \\ \beta \cdot (T_w - \gamma) & (\gamma < T_w < \frac{1}{\beta} + \gamma) \\ 1 & (T_w \geq \frac{1}{\beta} + \gamma) \end{cases} \qquad [9]$$

$\gamma$ is the limit of the tolerable waiting time in which a customer is willing to wait; $\beta$ is the customer loss percentage per unit time ($0 \leq \beta \leq 1$) such that the higher the $\beta$, more impatient customers are in the pool.

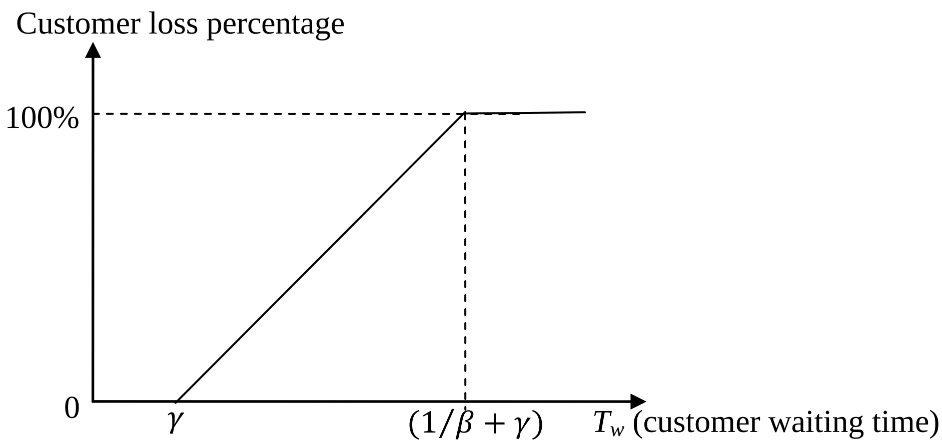



**Figure 3-7. Relationship of customer loss percentage against waiting time**

## 3.3 Determination of Production Ahead Time ($T_a$)

Now, recalling the governing function is:

$$\boxed{\begin{aligned} Z(T_a) = \rho\left[P_f(T_a)^+ + P_c(T_w)\right] \cdot L_o + (1-\rho) \cdot P_f(T_a)^- \cdot L_u \\ +(1-\rho) \cdot Max\left[P_c(T_w) - P_f(T_a)^-, 0\right] \cdot L_o \end{aligned}} \qquad [10]$$

Here, the assumption is made that the forecast error value at time $T_a$ has an even distribution in the error range and so the mean of the forecast error is half of the maximum value ($0.5\left(1-e^{-(\alpha\cdot T_a)}\right)$). Now, by inserting the forecast error percentage and customer loss parentage functions, then:

$$c_1 = L_o\left(0.5(1-e^{-(\alpha\cdot T_a)}) + \beta(T - T_a - \gamma)\right) \quad [11]$$

**Situation A: $\boldsymbol{P_c(T_w) > P_f(T_a)}$:**

$$c_2 = P_f(T_a)^- \cdot L_u + \left[P_c(T_w) - P_f(T_a)^-\right]\cdot L_o \quad [12]$$

$$= 0.5\left(1-e^{-(\alpha\cdot T_a)}\right)\cdot L_u + \left[0.5(e^{-(\alpha\cdot T_a)} - 1) + \beta(T - T_a - \gamma)\right]\cdot L_o$$

Hence, the function becomes:

$$Z(T_a) = \rho\left(0.5(1-e^{-(\alpha\cdot T_a)}) + \beta(T - T_a - \gamma)\right)\cdot L_o$$
$$+ (1-\rho)\left\{0.5\left(1-e^{-(\alpha\cdot T_a)}\right)\cdot L_u + \left[0.5(e^{-(\alpha\cdot T_a)} - 1) + \beta(T - T_a - \gamma)\right]\right.$$
$$\left.\cdot L_o\right\}$$

Then, by taking $\rho = 0.5$ (i.e., under-forecast and over-forecast have equal chances) and simplifying the above objective function, gives:

$$\boxed{Z(T_a) = 0.25\left(1-e^{-(\alpha\cdot T_a)}\right)\cdot L_u + \beta(T - T_a - \gamma)\cdot L_o} \quad [13]$$

Here, taking the derivative with respect to $T_a$,

$$Z(T_a)' = 0.25\alpha e^{-(\alpha\cdot T_a)}\cdot L_u - \beta L_o$$

When $Z(T_a)' = 0$, $T_a = \frac{1}{\alpha}\ln\frac{\alpha L_u}{4\beta L_o}$.

Taking the second derivative of the function,

$$Z(T_a)'' = -0.25\alpha^2 e^{-(\alpha\cdot T_a)} X L_u\ ,$$

Since $Z(T_a)'' < 0$ (one can see that it is always a negative number), then $Z(T_a)' = 0$ is the maximum point. Thus, $T_a = \frac{1}{\alpha}\ln\frac{\alpha L_u}{4\beta L_o}$ gives the maximum value in

**Situation A** and the curve shape of $Z(T_a)$ is shown in **Figure 3-8**.

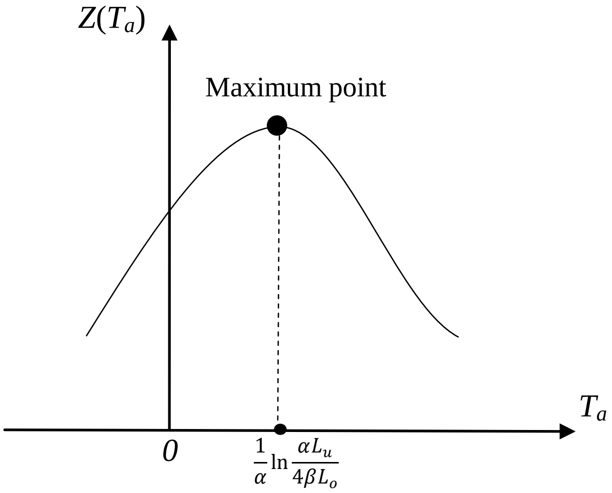



**Figure 3-8. $Z(T_a)$ curve in Situation A** $[P_c(T_w) > P_f(T_a)]$

**<u>Situation B: $P_c(T_w) < P_f(T_a)$:</u>**

$$c_2 = P_f(T_a)^- \cdot L_u$$

$$= 0.5[1 - e^{-(\alpha \cdot T_a)}] \cdot L_u \qquad [14]$$

The objective function becomes:

$$\boxed{\begin{aligned} Z(T_a) = 0.5\left(0.5\left(1 - e^{-(\alpha \cdot T_a)}\right) + \beta(T - T_a - \gamma)\right) \cdot L_o \\ +0.25\left(1 - e^{-(\alpha \cdot T_a)}\right) \cdot L_u \end{aligned}} \qquad [15]$$

Similarly, taking derivatives with respect to $T_a$,

$$Z(T_a)' = 0.5\left(0.5\alpha e^{-(\alpha \cdot T_a)} - \beta\right) \cdot L_o + 0.25\left(\alpha e^{-(\alpha \cdot T_a)}\right) \cdot L_u$$

$$= 0.25\alpha e^{-(\alpha \cdot T_a)} \cdot L_o - 0.5\beta L_o + 0.25\left(\alpha e^{-(\alpha \cdot T_a)}\right) \cdot L_u$$

$$= 0.25\alpha e^{-(\alpha \cdot T_a)}(L_u + L_o) - 0.5\beta L_o$$

Since $Z(T_a)'' = -0.25\alpha^2 e^{-(\alpha \cdot T_a)} X \cdot (L_u + L_o) < 0$ (negative value), $Z(T_a)' = 0$ is also the maximum point. Therefore, $T_a = \frac{1}{\alpha}\ln\frac{\alpha}{2\beta}(1 + \frac{L_u}{L_0})$ is the maximum value in **Situation B** and the sketched curve of $Z(T_a)$ is given in **Figure 3-9**.

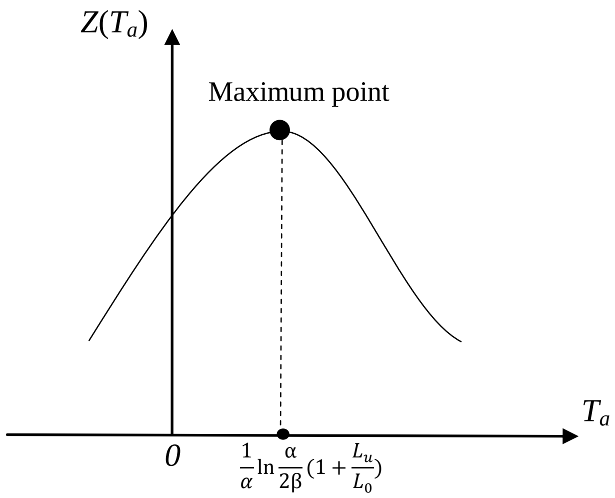



**Figure 3-9. $Z(T_a)$ curve shape in Situation B** $[P_c(T_w) < P_f(T_a)]$

In fact, there is little interest in these two maximum points as each represents one of the worst scenarios, but the goal is clearly on minimizing the loss by identifying an appropriate $T_a$ (the production start ahead time) within the range [0, T]. To visualize the two $Z(T_a)$ curves governing situations A and B, they have been sketched in **Figure 3-10** (a, b and c) according to different circumstances.

It is not difficult to observe that the intercept of these two curves occurs at $P_f(T_a) = P_c(T_w)$; also see the formulation of the equation [3] for $C_2$. In the figures, the minimum value can be found at three points:

$$T_a = 0 \qquad [16]$$

$$T_a = (T - \gamma) \qquad [17]$$

$$P_f(T_a) = P_c(T_w) \qquad [18]$$

in which the condition of $P_f(T_a) = P_c(T_w)$ is particularly fascinating.

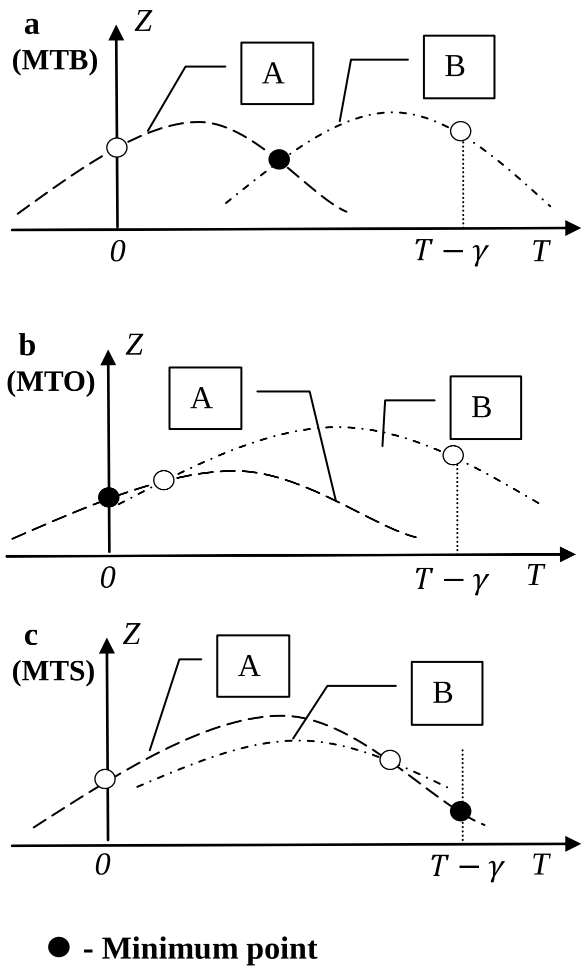



**Figure 3-10. $Z(T_a)$ curves in MTB/MTS/MTO**

In the case of $P_f(T_a) = P_c(T_w)$,

$$P_f(T_a) - P_c(T_w) = 0$$

This means that, with this production start ahead time ($T_a$), the products are the right fit to the customer demand after waiting for ($T_w$), as the quantity of underproduction is equal to the customer loss in waiting. See **Figure 3-11**.

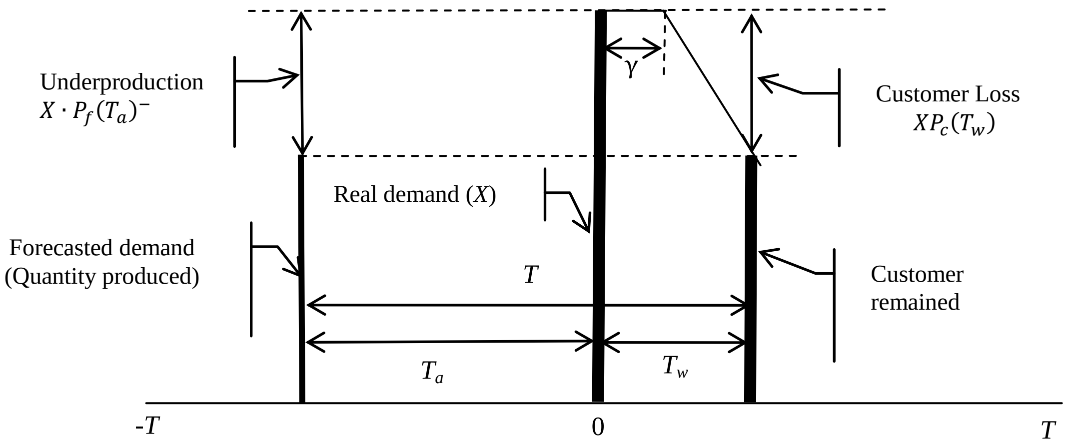



**Figure 3-11. Underproduction equal to customer loss**

By inserting the associated functions developed with the condition of $P_f(T_a)$ equals to $P_c(T_w)$, gives:

$$(1 - e^{-(\alpha \cdot T_a)}) - \beta(T - T_a - \gamma) = 0 \qquad [19]$$

As the value of $\alpha T_a$ is usually quite small in order to provide a reasonably good forecast result (say, for a forecast error of ±20% in a 100-day time frame $(1 - e^{-(100\alpha)} = 20\%)$, it can be readily worked out that $\alpha=0.0022$, $\alpha T_a$=0.2200, $(\alpha T_a)^2$=0.0484, $(\alpha T_a)^3$=0.0106), etc. Since the exponential function can also be expanded as $e^{-x} = \sum_{n=0}^{\infty}(-1)^n \frac{x^n}{n!} \cong 1 + \frac{-x}{1!} + \frac{(-x)^2}{2!} + \frac{(-x)^3}{3!} + \cdots$, it provides a sensible approximation by ignoring the third order term and above in the equation. Thus,

$$1 - \left[1 - \alpha T_a + \frac{(\alpha T_a)^2}{2}\right] - \beta(T - T_a - \gamma) \cong 0$$

$$0.5(\alpha T_a)^2 - (\alpha + \beta)T_a + \beta(T - \gamma) \cong 0$$

Therefore, the $T_a$ at the interception point ($T_{ax}$) can be resolved by calculating the roots of the quadratic equation:

$$\boxed{T_{ax} \cong \frac{\alpha+\beta \pm \sqrt{(\alpha+\beta)^2 - 2\alpha^2\beta(T-\gamma)}}{\alpha^2}} \qquad [20]$$

Hence, through the comparison of $T_a$ at this interception and the two boundary points, a decision can be made with the help of the function by minimizing:

$$Z(T_a) = \rho\left[P_f(T_a)^+ + P_c(T_w)\right] \cdot L_o + (1-\rho) \cdot P_f(T_a)^- \cdot L_u$$

$$+(1-\rho) \cdot Max\left[P_c(T_w) - P_f(T_a)^-, 0\right] \cdot L_o$$

For $\alpha = 0$ (totally accurate forecast), the optimal value can be found at $T_a = T - \gamma$ and it is an MTS case. When $\beta = 0$ (very good, patient customers), $\gamma = T$ (customers are willing to wait for a total production time) or $\alpha = \infty$ (the forecast is totally inaccurate), the optimal value is at $T_a = 0$ that is MTO. Hence, to facilitate the decision-making process, the optimal judgment conditions are summarized as:

For ($\alpha \cdot \beta \neq 0$:), one has to check the $T_a$ at all three points as:

$$Min\, Z(T_a) = Min\left[Z(0),\ Z\left(\frac{\alpha+\beta-\sqrt{(\alpha+\beta)^2-2\alpha^2\beta(T-\gamma)}}{\alpha^2}\right),\ Z(T-\gamma)\right];$$

For ($\alpha = 0$) or ($\beta = 1$ and $\gamma = 0$):

The optimal value is at $T_a = T - \gamma$;

For ($\beta = 0$), ($\gamma = T$) or ($\alpha = \infty$):

The optimal value is at $T_a = 0$.

As a supplement, there is another quick approximation by ignoring the second order term and those beyond in the $e^{-x}$ equation. Thus,

$$1 - [1 - \alpha T_a] - \beta(T - T_a - \gamma) \cong 0$$

$$\boxed{T_{ax} \cong \frac{\beta(T-\gamma)}{(\alpha+\beta)}} \qquad [21]$$

Then, when $\alpha \cdot \beta \neq 0$, the optimal judgment conditions can be replaced by:

$$Min\, Z(T_a) = Min\left[Z(0),\ Z\left(\frac{\beta(T-\gamma)}{(\alpha+\beta)}\right),\ Z(T-\gamma)\right]$$

## 3.4 Summary

MTB framework and mathematical model were presented in this chapter. The main idea of MTB is to achieve minimum production loss through the balancing of the customer loss in waiting and the forecast error. First, the forecast error spread percentage function and customer loss in waiting percentage function were introduced. Next, two situations (overproduction and underproduction) were elaborated. Then, based on these elements, the objective function of MTB was formulated to determine the most suitable production starting time that would be sometime between the MTO and MTS time horizon; in some occasions, the optimal solution could be either MTO or MTS as well because the result depended on system parameters. Finally, a simplified version of the MTB model was also presented for easing the calculated.

# Chapter 4 MTB Software and Simulation Model Developments

This chapter describes the simulation model developed by using the STELLA software and C# language to identify the correctness of the proposed MTB model. As a commercially available software package, STELLA offers an opportunity to create dynamic visual models for studying a wide variety of problems in an easy to understand style. However, it can be tedious to use STELLA to generate all necessary data for verification purposes and, therefore, the program MTB has been coded. It uses a step-by-step iteration approach similar to the simulation and the correctness is confirmed by STELLA. Last, the Smart MTB has also been created to determine the most suitable "production start ahead time" based on the proposed MTB model straightaway.

## 4.1 STELLA Simulation Model

The main STELLA window is divided into four tabbed pages: Interface, Map, Model and Equation. Each tab represents a distinct layer in the model and provides a different way for designing and presenting a model. This section illustrates details of how the STELLA model simulates the results of different production arrangements constructed.

### 4.1.1 Simulation Interface

The interface of the STELLA model is composed of two parts, the *Parameter Input List* and the *Simulation Result* table that shows the total loss per product as in **Figure 4-1**. In the parameter input list, there are nine parameters: *Forecast Demand Qty, Forecast Error Spread % at Total Time Required Ahead, Customer Loss Rate,* *Customer Willing to Wait Time, Delivery Time, Production Rate, Product Throughput Time, Overproduction Cost Per Unit* and *Underproduction Cost Per Unit*. Next, the *Simulation Result table* shows the results of total loss per product in the simulation process with different $T_a$ settings.

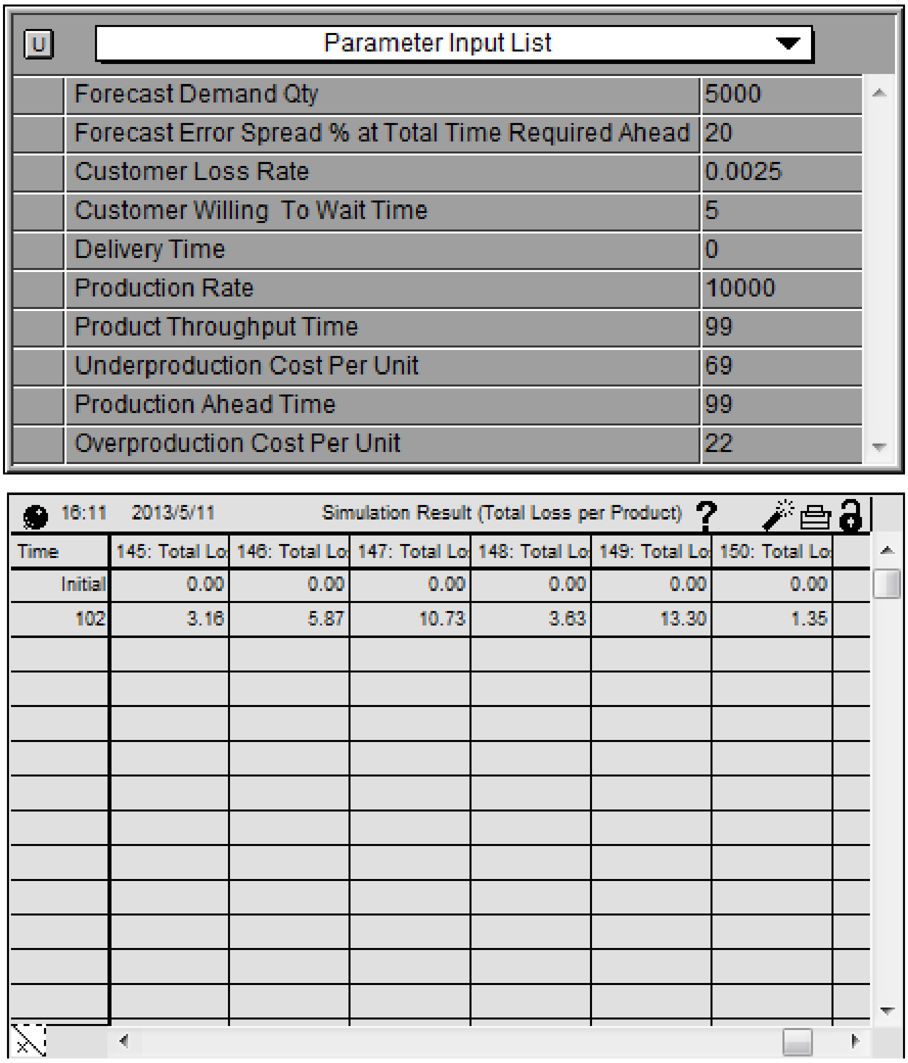



**Figure 4-1. STELLA model interface**

### 4.1.2 Simulation Sub-models

With the forecast demand to be produced, the known production rate, throughput time and delivery time needed and the total time ($T$) required from the production start to delivery of products to customers can be determined. **Figure 4-2** shows the inter-links, and details about the associated mathematical relationships can be found in **Table 4-1**.

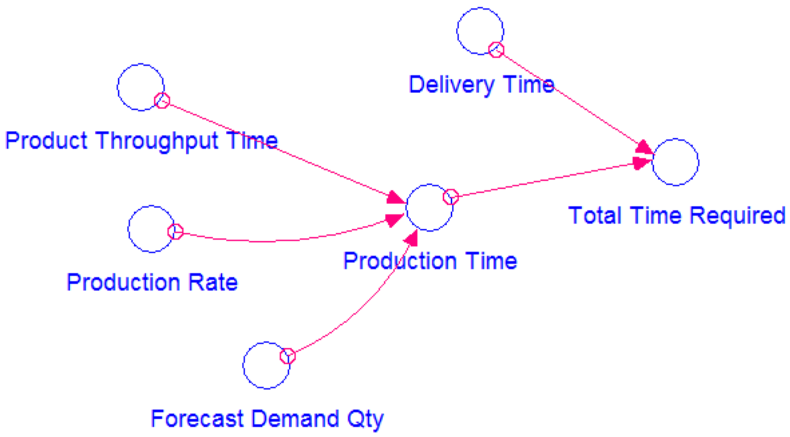



**Figure 4-2. Simulation process of total time calculation**

As the forecast demand is the quantity to be produced, once the forecast error at $T_a$ is known, the range of customer demand after time $T_a$ can be computed accordingly. By making use of the random generator in the software with event distribution in the given customer demand range, a customer demand can be created. On the other hand, the customer waiting time can be obtained simply ($T_w = T - T_a$). Through the simulation, possible inventory and products sold can be obtained at each $T_a$, and the total loss per product can be obtained in each case. Next, the collective simulation results give the mean value of the total loss per product for a particular $T_a$. Last, keep trying incrementally for next $T_a$ value, and the most suitable production ahead time ($T_a$) can be found accordingly.

In practice, there are five main clusters for the simulating production. The first is to calculate the total time needed for producing the forecast product quantity, as in **Figure 4-2**. The second is to simulate the customer demand. With the forecast error spread and total time required, the forecast error factor (α) can be determined. Then, with the production given ahead time $T_a$ and the forecast demand quantity, the forecast error spread at $T_a$ can be found. In the range of ±forecast error spread, a random number (Event Distributed Random Number) is generated to simulate the "real" forecast error so that the "real" demand quantity can be calculated. The simulation process is shown in **Figure 4-3**.

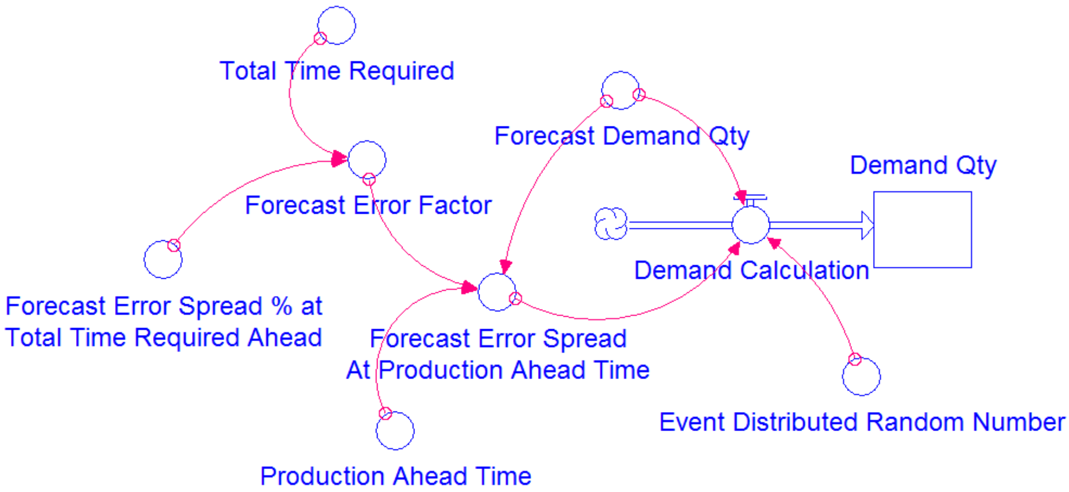



**Figure 4-3. Simulation of real customer demand**

The third cluster involves working out the customer loss in waiting. With the customer waiting time ($T_w$) and the two customer loss percentage factors ($\gamma$: the limit of the tolerable waiting time; $\beta$: the customer loss percentage per unit time), customers still remaining after a waiting time ($T_w$) can be resolved. **Figure 4-4** shows a STELLA model cluster for the demand, after a waiting that is equivalent to the number of customers remaining in the pool at that time.

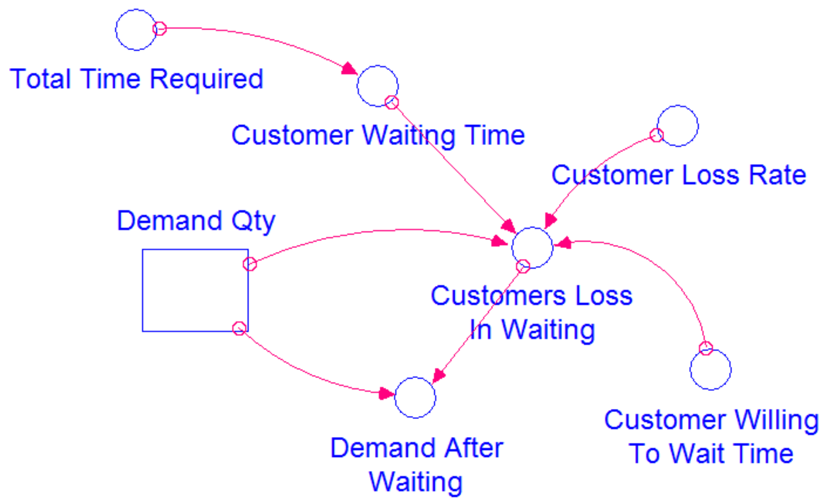



**Figure 4-4. Simulation of customer loss in waiting**

The fourth cluster is the simulation of the production flow. It starts by producing the forecast demand quantity and the products that are available after the production time. The products delivered to customers are marked as products sold and the products left stay in the inventory. This simulation cluster is shown in **Figure 4-5**.

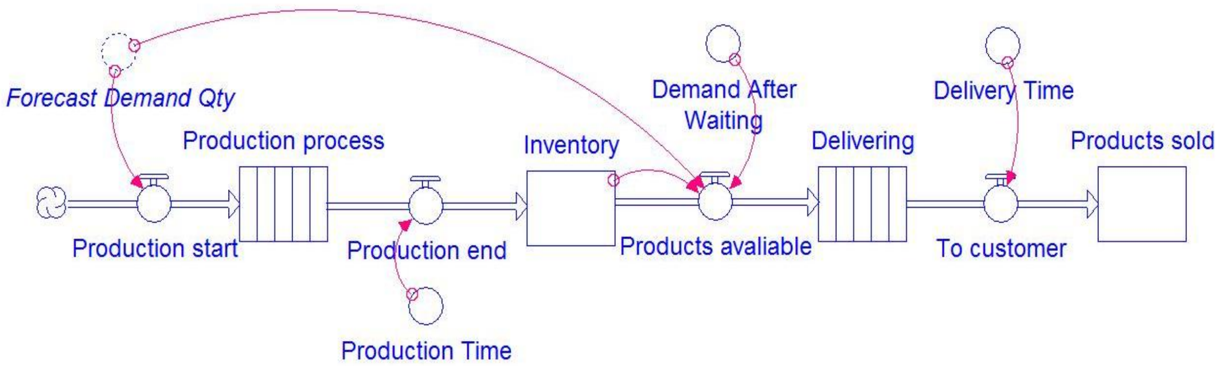



**Figure 4-5. Simulation of production and delivery**

The fifth cluster is to work out the underproduction loss or overproduction loss based on the information provided in previous clusters so that the total loss and total loss per product can be determined. (Also see **Figure 4-6**)

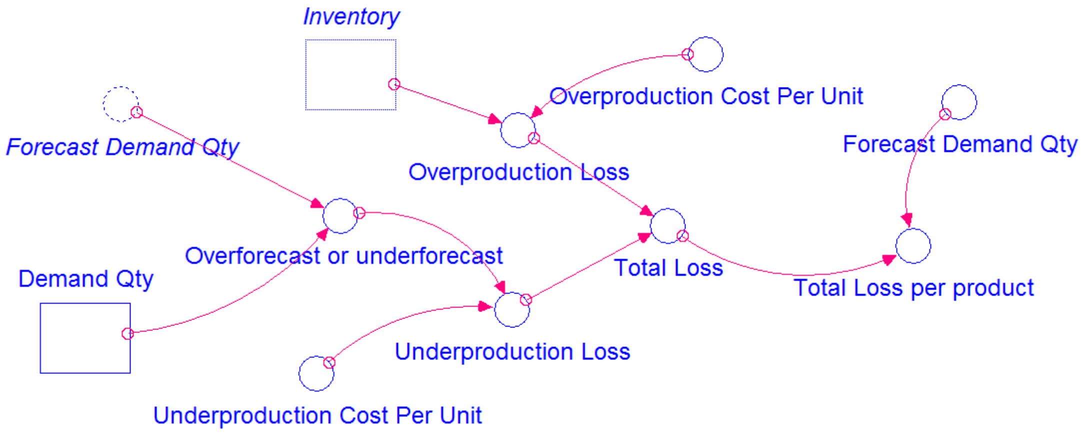



**Figure 4-6. Simulation of loss calculation**

**Figure 4-7** shows the entire simulation model that contains the five clusters illustrated, and **Table 4-1** summarizes all the main equations involved in this simulation model.

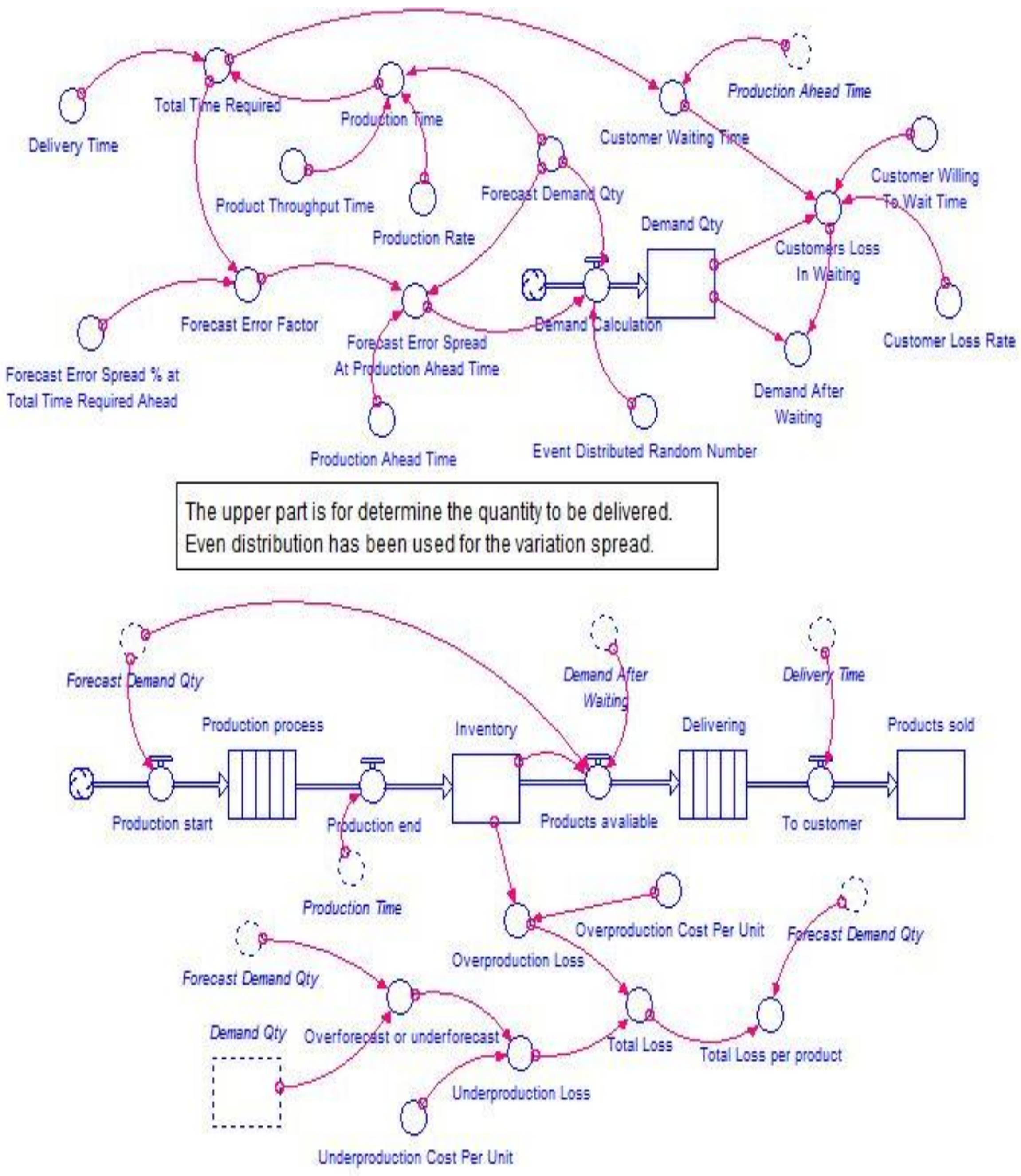



**Figure 4-7. MTB STELLA simulation model**

**Table 4-1. Equations of MTB simulation model in STELLA**

| Parameter | Equation |
|---|---|
| Product time | ROUND(Product_Throughput_Time+Forecast_Demand_Qty/Production_Rate) |
| Total time required | Delivery_Time+Production_Time |
| Forecast error factor | -1*LOGN(1-Forecast_Error_Spread_%_at_Total_Time_Required_Ahead/100)/Total_Time_Required |
| Forecast Error Spread at Production Ahead Time | ROUND((1-EXP(-1*Forecast_Error_Factor*Production_Ahead_Time))*Forecast_Demand_Qty) |
| Demand Calculation | PULSE(Forecast_Demand_Qty+Forecast_Error_Spread_At_Production_Ahead_Time*Event_Distributed_Random_Number, 0, 0) |
| Customer Waiting Time | Total_Time_Required-Production_Ahead_Time |
| Customers Loss in Waiting | IF Customer_Willing__To_Wait_Time >= Customer_Waiting_Time Then 0 Else IF (Customer_Waiting_Time-Customer_Willing__To_Wait_Time)*Customer_Loss_Rate >= 1 THEN Demand_Qty ELSE Demand_Qty*(Customer_Waiting_Time-Customer_Willing__To_Wait_Time)*Customer_Loss_Rate |
| Demand After Waiting | Demand_Qty-Customers_Loss__In_Waiting |
| Production start | PULSE(Forecast_Demand_Qty,0,0) |
| Production end | Production_Time |
| Products avaliable | IF Inventory = Forecast_Demand_Qty THEN MIN(Inventory,Demand_After__Waiting) ELSE 0 |
| Overproduction Loss | Inventory*Overproduction_Cost_Per_Unit |
| Overforecast or underforecast | Forecast_Demand_Qty-Demand_Qty |
| Underproduction Loss | IF Overforecast_or_underforecast < 0 THEN -Overforecast_or_underforecast*Underproduction_Cost_Per_Unit ELSE 0 |
| Total Loss | Overforecast_Loss+Underforecast_Loss |
| Total Loss per product | Total_Loss/Forecast_Demand_Qty |

### 4.1.3 Simulation Model Configuration

**Figure 4-8** shows the 'Run Specs' configuration of a STELLA simulation model. The two circled portions are where special attention is needed. The 'length of simulation' is the model running time that should always be larger than the total production time required in the case of simulating the MTB model. 'DT' is the time increment for calculations in a model simulation.

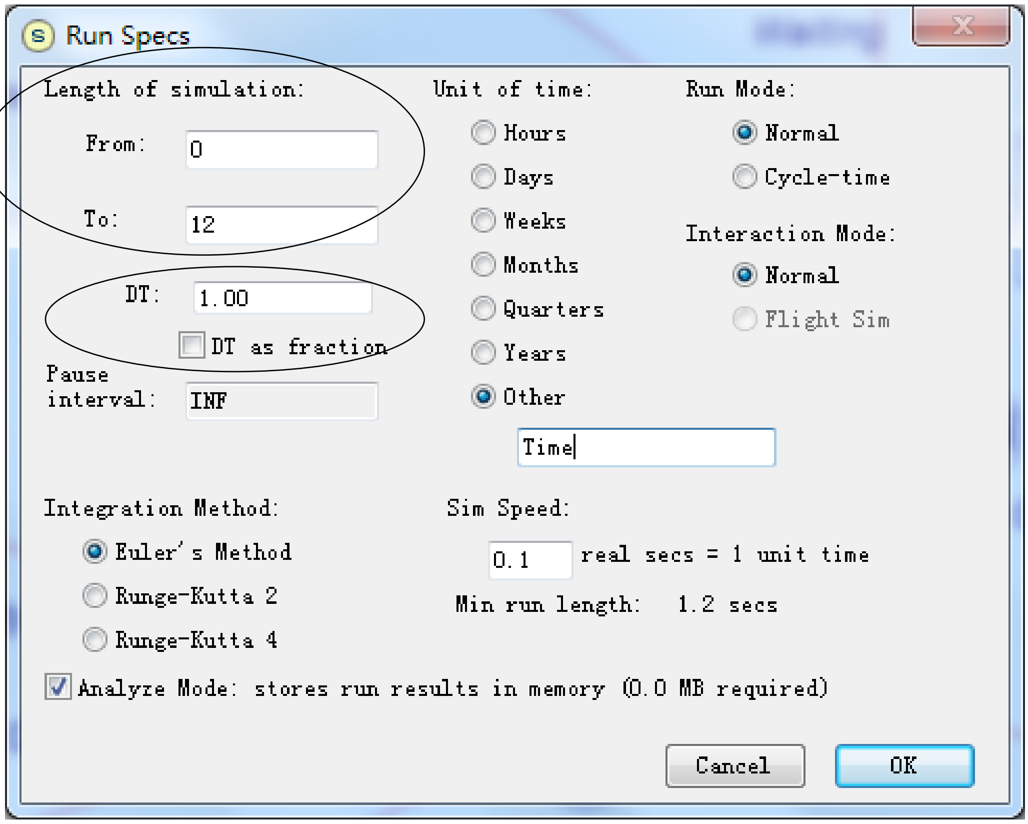



**Figure 4-8. 'Run Specs' configuration of MTB model in STELLA**

STELLA provides the sensitivity analysis function that can be used to generate a group of simulation results. This is very useful for collecting statistical data and can be employed to obtain the average total loss per product in the MTB model analysis because the “real” demand in simulation will be generated based on random distribution. For example, in **Figure 4-9**, the value of '# of Runs' is set to 150 and means that the test will run 150 times automatically, each time with a random number evenly distributed (see the icon on the left hand side of the distribution checkbox selected) in the range of [-1, 1] and a seed equal to 0. Finally, the outputs can be stored in a table and one can work out the mean value needed accordingly by transfer of the output data into the MS-Excel format.

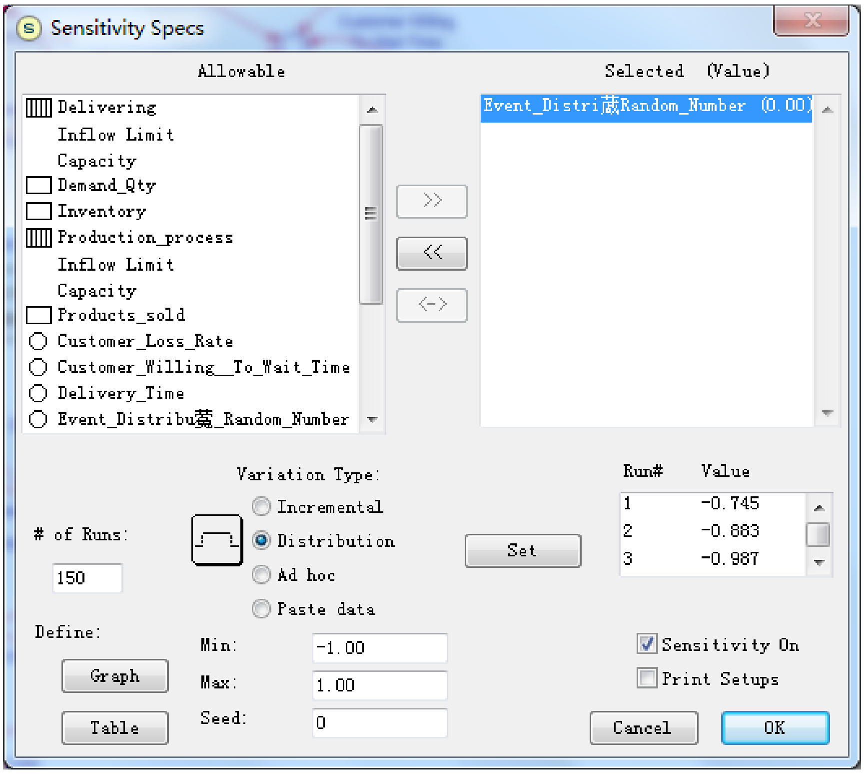



**Figure 4-9. Sensitivity configuration in STELLA**

## 4.2 Introduction to Two C# MTB Programs

In this research, two programs were designed and coded in C# on the Microsoft Visual C# 2008 Express Edition platform. The MTB program gives the step-by-step iteration results based on incrementing the production start ahead times by: (1) using random numbers to generate the work, like the simulation approach, and (2) utilizing a constant value $P_f(T_a)/2$ to represent the forecast error for ease of calculation, and this also confirms the accuracy of putting a 0.5 value in **Equation**

**[10]** in Chapter 3. This program plots the loss per product against production start ahead time, so that one can easily observe the minimum on the graph as well. The aim is to verify the correctness of the proposed MTB model. Although STELLA can also have the job done, it consumes much more time and effort in getting the required results and MS-Excel is needed to assist in the data manipulations. The program architecture, flow logic, functions and interface are presented in the following sections. The experimental data are inputted through the user interface where some essential outputs are also shown after the program execution. In addition, the program execution results are also stored in an MS Excel file in “csv” format for viewing. Finally, the smart version of the program, namely Smart MTB, allows the obtaining of the best production start ahead time and its linked cost loss per product directly by employing **Equation [20]**.

### 4.2.1 MTB Software Architecture

The main purpose of coding this software program in this research was to verify the correctness of the generated results by the formulated mathematical model. **Figure 4-10** gives a conceptual picture of the framework with some crucial data connections. The *Production Ahead Time Determination* module, which is the kernel of the software, contains the equation set with the operational sequences established in line with **Chapter 3**. To determine the consequences of having different production ahead times, basic setups of the forecast model and the customer behavior to the associated product need to be done beforehand. Then, with the product information (total production duration time, overproduction loss and underproduction loss), the most suitable production ahead time ($T_a$) can be obtained by tracking for the lowest total loss per product with incrementing production ahead time setting, on top of using the proposed MTB model created.

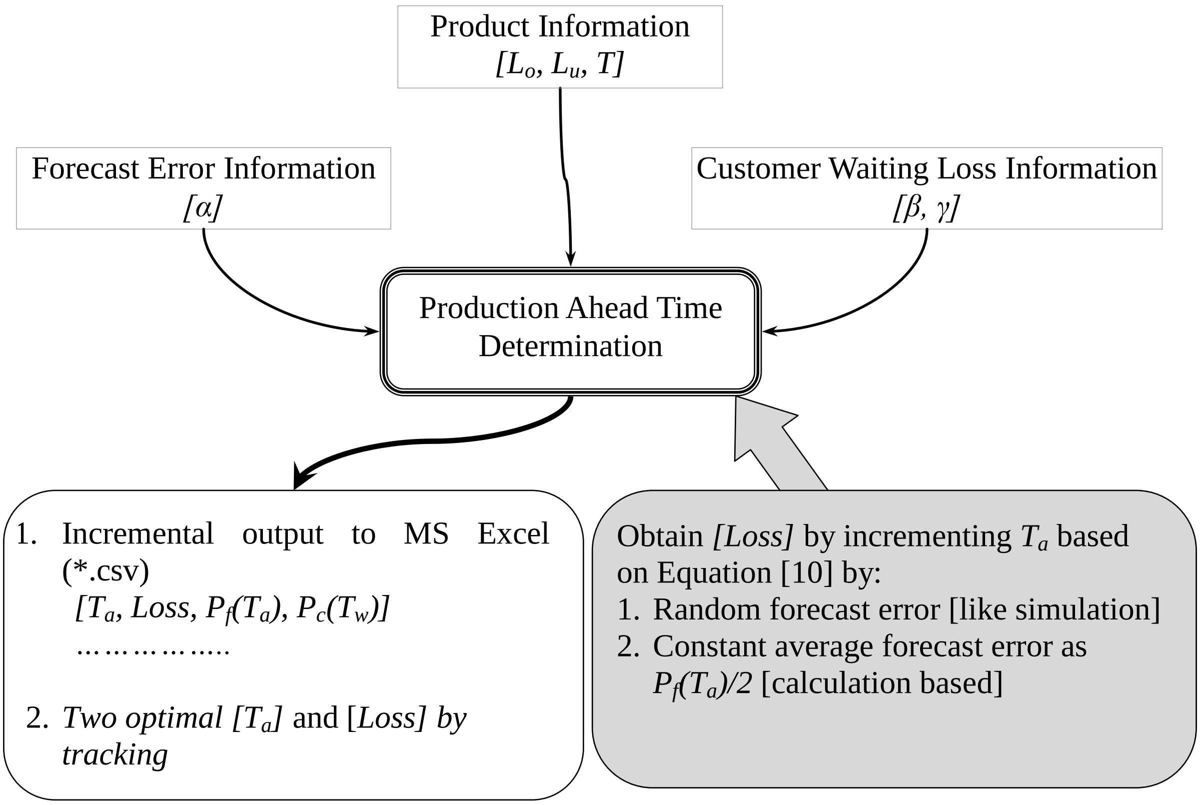



**Figure 4-10. MTB program conceptual framework**

### 4.2.2 MTB Program Workflow and Key Functions

With reference to **Figure 4-10**, it can be seen that the implementation of the production ahead time determination module is the most important portion of the program. To enhance the user-friendliness of the MTB program, it is separated into three parts on the window: the numeric data entry, the numerical result display and the graphical drawings. The sequence diagram regarding the sofware operations is shown in **Figure 4-11**. The first part concerns the ***Form Generation***, whereas the second part labelled as ***MTB Calculations*** is the core which contains seven key functions:

```
void ValueAssign() //assign the value to all parameters
double ForecastError(int ta) //calculate forecast error
double CustomerLoss(int tw) //calcualte percentage of customer loss
double Loss(int ta) //simculate loss at time (ta)
double CalLoss(int ta) //iterative calculate loss at time (ta)
void DataToExcel() //write calculation result to excel file
void MarkUpandLowBound() //find the Max and Min value of loss
void CalculateOptimumTa() //calculate optimal Ta
```

**Figure 4-12** provides the operation detail for these seven core functions. The third part in **Figure 4-12** is the ***Curve Drawing***. It has six functions:

```
private void panel1_Paint(object sender, PaintEventArgs e)
private void Panel1_MouseClick(object sender, MouseEventArgs e) //draw forecast error curve
private void panel2_Paint(object sender, PaintEventArgs e)
private void panel2_MouseClick(object sender, MouseEventArgs e) //draw customer loss curve
private void Panel3_Paint(object sender, PaintEventArgs e)
private void Panel3_MouseClick(object sender, MouseEventArgs e) //draw total loss curve
```

The flowchart of the ***Curve Drawing*** is in **Figure 4-13**. In the following sections, only essential areas are illustrated and the skeletons of some functions are presented. The program source codes are provided in **Appendix-A** for reference. Corresponding symbols used in **Chapter 3** are shown in round brackets in the program skeleton.

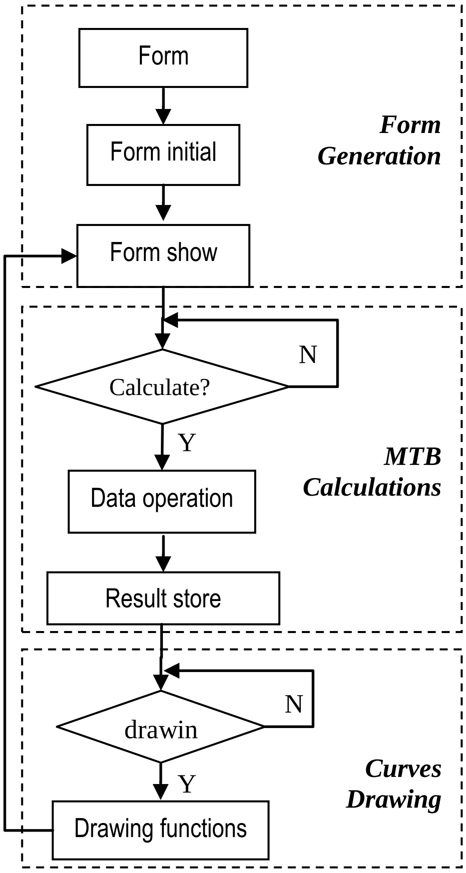



**Figure 4-11. MTB program flowchart**

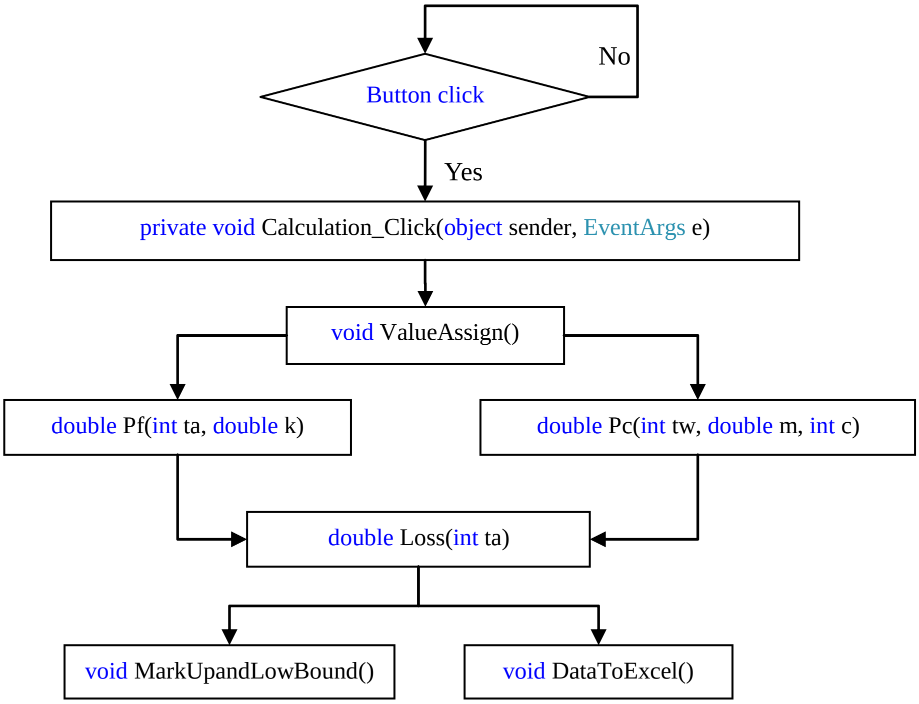



**Figure 4-12. MTB calculations flowchart**

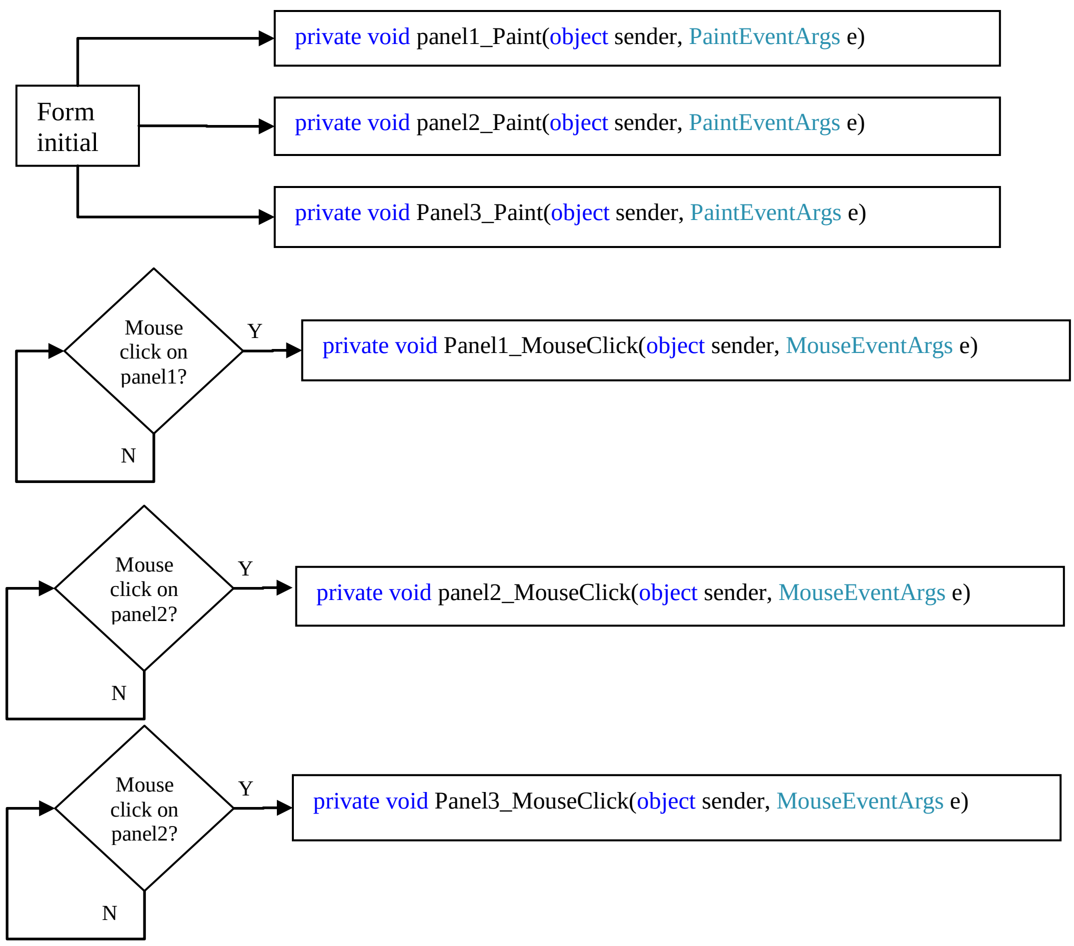



**Figure 4-13. Curves drawing flowchart**

### 4.2.3 MTB Program Development

First is the ***Form Generation***. It uses a standard style of creating a form in the Widnow environment. The *InitializeComponent* function is to invoke all associated form components while the *Initial* function is to initialize variables used in the program and to provide memory spaces for them.

```
void InitializeComponent ();
        /** initialize interfacce form and prepare memory     */

void Initial ();
        /** init extra memory for varibles used in following calculations   */
```

### 4.2.4 MTB Calculation Functions

The *ValueAssign* is for basic parameter setups such as the throughput time and the product related information, etc. *ForecastError* calculates the forecast error and *CustomerLoss* determine the customer loss percentage after waiting for a time $T_w$. *Loss* works out the total loss in a production ahead time $T_a$. *DataToExcel* and *DataToMatrix* provide output data to a MS-Execl file and to a matrix respectively, for later operations. *Calculation_Click* is to execute the calculations by calling the mentioned functions.

```
void ValueAssign()
        /** parameter value assign          */
```

```
double ForecastError(int ta)
        /** calculation forecast error
        @param:
                ta                              forecast ahead time
        @return
                1 − e^(−α·ta)                   forecast error
        */
```

```
double Loss(int ta)
        /** total Loss
        @param:
                ta                              forecast ahead time
        @return
                Z                               Total   profit loss
        */
```

```
double CalLoss(int ta)
        /** calculation total Loss
        @param:
                ta                              forecast ahead time
        @return
                Z                               Total   profit loss
        */
```

```
double CustomerLoss(int tw, double m, int c)
        /** setup memory size for propable events
        @param:
                        tw                                      customer waiting time
                        m                                       customer loss percentage per unit time
                        c                                       maximum waiting time customer can tolerate
            @return
                        m(tw-c)                                 customer loss percentage
        */
```

```
void DataToExcel()
        /** write the result data to excel file     */

void MarkUpandLowBound()
        /** find the Max and Min value of loss  */

private void Calculation_Click(object sender, EventArgs e)
        /**answer the action of click "calculation" button
        @param:
                        sender                                  operate object
                        e                                       parameter of operation event
        @return
                        null
        */
```

### 4.2.5 Curve Drawing Functions

There are three curves shown on the interface window; the forecast error curve, the customer loss percentage curve and the loss at different $T_a$ curve. Six drawing functions are used to plot the afore mentioned curves. Three functions (namely *panel1_Paint*, *panel2_Paint* and *panel3_Paint*) are utilized to produce the X and Y axes for these three curves. Once the calculation data are available, another three drawing functions (*Panel1_MouseClick, Panel2_MouseClick, Panel3_MouseClick*) are employed to execute the drawing actions.

```
  private void panel1_Paint(object sender, PaintEventArgs e);
        /** drawing the X and Y axial of foecast accuracy curve on panel 1 while form initialization
        @param
                        sender                                  operate object
                                e                                       parameter of paint operation event
        @return
                        null
        */
```

```
private void Panel1_MouseClick(object sender, MouseEventArgs e);
        /** drawing the X and Y axial and update the drawing of foecast accuracy curve on panel 1 while mouse click
the drawing area
        @param:
                        sender                          operate object
                        e                               parameter of mouse operate event
        @return
                        null
        */
```

```
  private void panel2_Paint(object sender, PaintEventArgs e);
            /** drawing the X and Y axial of customer loss percentage curve on panel 2 while form initialization
        @param:
                        sender                          operate object
                        e                               parameter of paint operate event
        @return
                        null
        */
```

```
private void Panel3_Paint(object sender, PaintEventArgs e);
        /** drawing the X and Y axial of total profit curve on panel three on panel 3 while form initialization
       @param:
                        sender                          operate object
                       e                                        parameter of paint operate event
        @return
                        null
        */
```

```
private void Panel3_MouseClick(object sender, MouseEventArgs e);
        /** drawing the X and Y axial and update the drawing of total profit curve on panel 3 while mouse click the
drawing area
        @param:
                        sender                          operate object
                        e                               parameter of mouse operate event
        @return
                        null
        */
```

```
private void panel2_MouseClick(object sender, MouseEventArgs e);
        /** drawing the X and Y axial and update the drawing of customer loss percentage curve on panel 2 while
mouse click the drawing area
        @param:
                                sender          operate object
                            e                   parameter of mouse operate event
        @return
                        null
            */
```

### 4.2.6 MTB Program Interface

**Figure 4-14** shows the MTB program interface. The upper left part is for the parameters input, and both the simulation and calculation results are provided at the lower left of the display window; the way in which the step-by-step increment of $T_a$ gets the optimal result works like simulation. The right side is for outputting the curves, forecast error, customer loss percentage and cost loss at different $T_a$. Through the comparison of the simulation and calculation results, the correctness and the optimal condition can be observed easily. In addition, the simulation is done by means of small step-progressive operations, and **Appendix-A** gives more information.

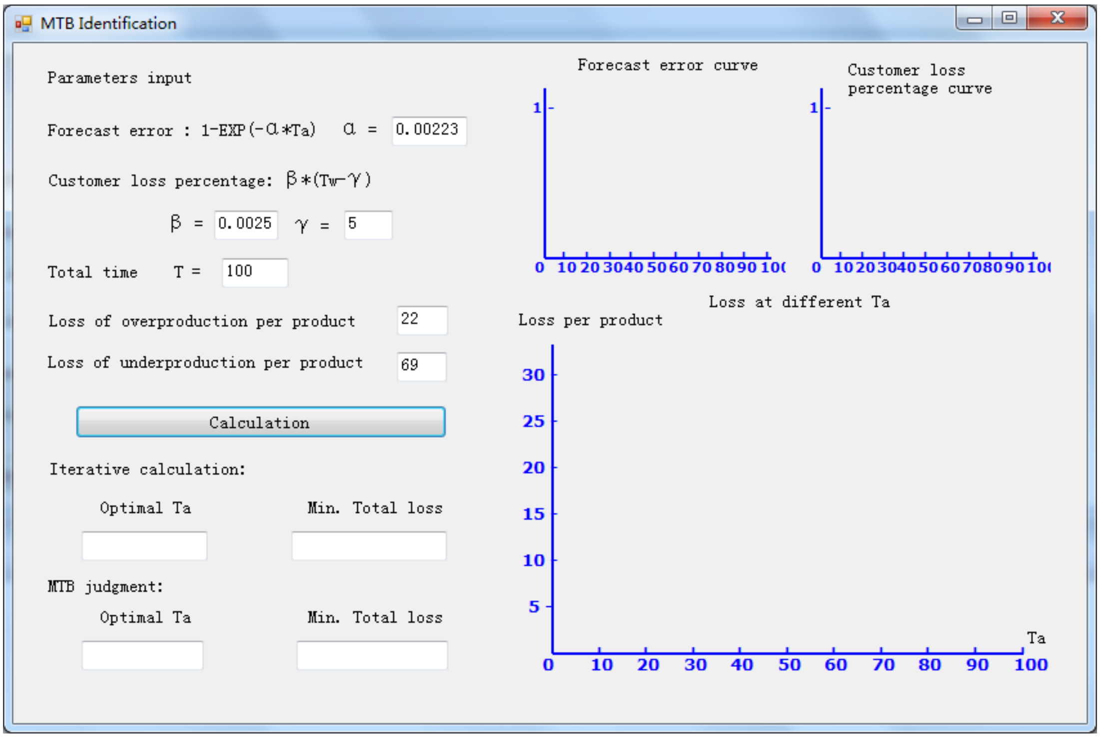



**Figure 4-14. MTB program interface**

## 4.3 Smart MTB Program

Apart from the previous MTB program, a smart version was also produced that calculates the optimal value of production ahead time ($T_a$) based on **Equation [20]**. It works differently in comparison with the MTB program and abandons those unnecessary parts, such as graph plotting, and incremental tracking. **Figure 4-15** shows the Smart MTB interface. This program merely provides fields for the parameter inputs and the optimal results. The program codes are given in **Appendix-B**.

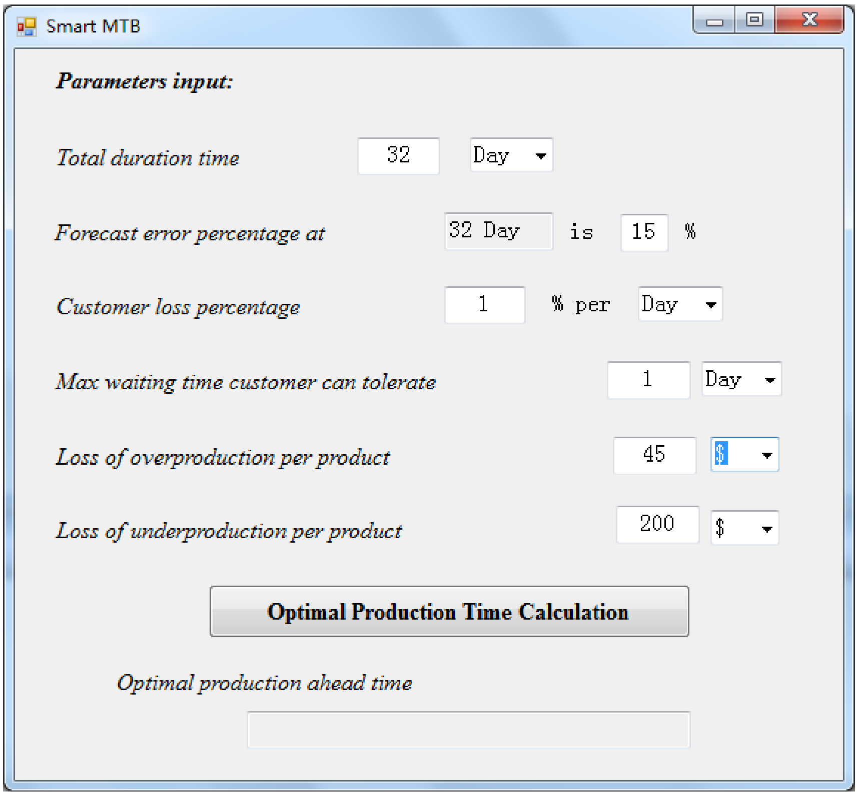



**Figure 4-15. Smart MTB calculation program interface**

# Chapter 5 Model Test and Date Analysis

The correctness of the proposed MTB model will be examined in this chapter, with not only the common situations addressed, but some extreme situations also considered in Sections 5.1 and 5.2. Analyses of the influences of the key parameters in the MTB model are described in Section 5.3. The computational accuracy of the optimal judgment based on the MTB model developed in **Chapter 3** was verified by "simulation" and these results are given in **Appendix-C, D, E, F** and **G**. In Section 5.5, the performance of MTB is evaluated and is also compared with both MTO and MTS.

## 5.1 MTB Model Testing

To demonstrate how the proposed MTB model works, a basic case has been set up. In this example, the total time duration ($T$) is 100 time units (e.g., days), and the forecast error spread has been estimated to be ±20% at $T_a = T$ (=100 time units) with the lowest forecast error value equal to 0 at $T_a = 0$ (i.e., MTO). The tolerable waiting time is 5 time units with no customer loss but there is a 0.25% customer loss per time unit afterwards. Last, the overproduction and underproduction losses are 22 and 69 cost/unit respectively. **Table 5-1** summarizes all the required details and the forecast error parameter (α) can be determined as:

$$P_f(T_a) = 1 - e^{-(\alpha \cdot T_a)}$$

$$0.2 = 1 - e^{-(100\alpha)}$$

$$\alpha = 0.00223$$

**Table 5-1. Forecast and customer loss test parameters**

| $T$ | $L_u$ | $L_o$ | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|---|
| 100 | 69 | 22 | 0.00223 | 0.0025 | 5 |

### 5.1.1 STELLA Simulation

Referring to **Table 5-1**, the STELLA simulation model Parameter Input List is configured as **Figure 5-1**, in which the Forecast Demand Qty is the expected sales volume. To make the simulation easier, it is assumed that the delivery time is zero (no delivery time is needed in this case), the production is stable and there are no accidental problems such as machine breakdowns. Therefore, the total required time ($T$ = throughput time + quantity/production rate + delivery time = 99 + 5000/10000 + 0 ≈ 100).

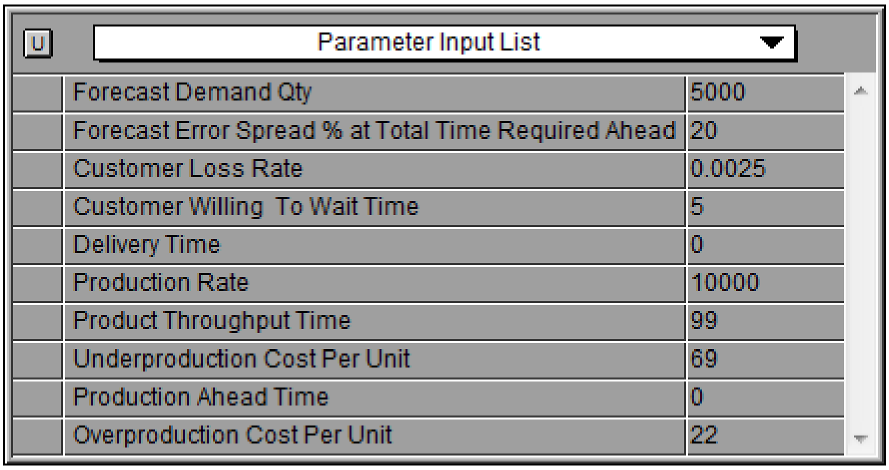



**Figure 5-1. STELLA simulation model parameter input**

The 'Run Specs' configuration is given in **Figure 5-2**. The length of the simulation was set to 103 time units, that is slightly larger than $T$ (=100) to cover the whole period. “DT” is 1 that means that here was one unit time interval in the simulation.

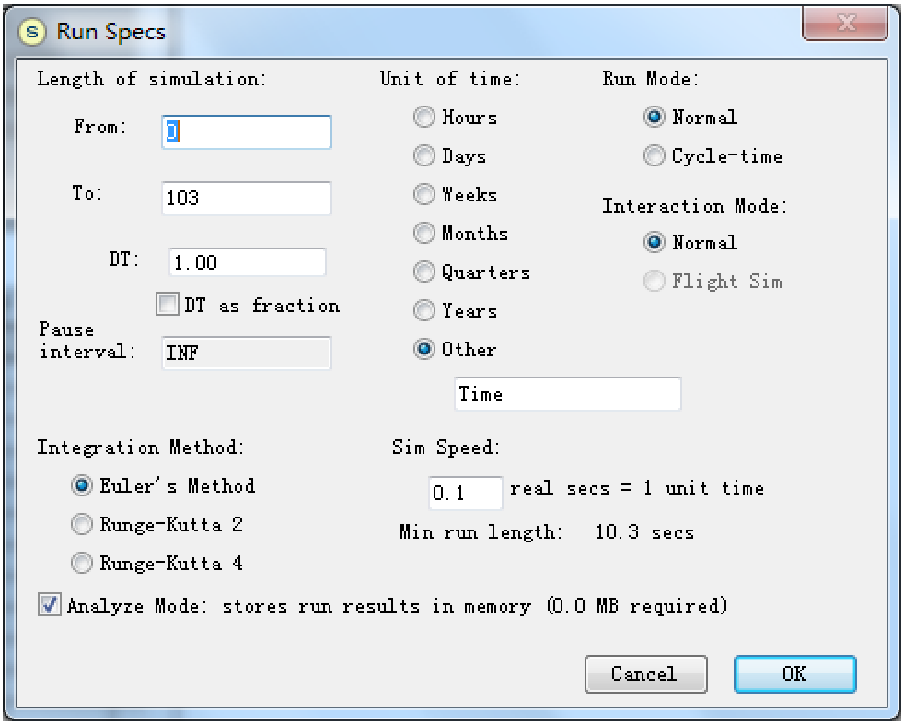



**Figure 5-2. The 'Run Specs' configuration of STELLA in MTB testing**

150 'total loss per product' values were obtained by running the STELLA model for each production ahead time ($T_a$) setting and the average value was calculated at each $T_a$. The $T_a$ started from 0 to 100 with an increment of 1 each time, so that the STELLA model was run 101 times. The result is plotted in graphical form as shown in **Figure 5-3**, with the details given in Appendix **C**. Indeed, this was quite a tedious operation and this is why we would like to make use of the MTB program to assist the work but, of course, STELLA offers better process visualization, as well as being easier to understand through the model building network reasoning.

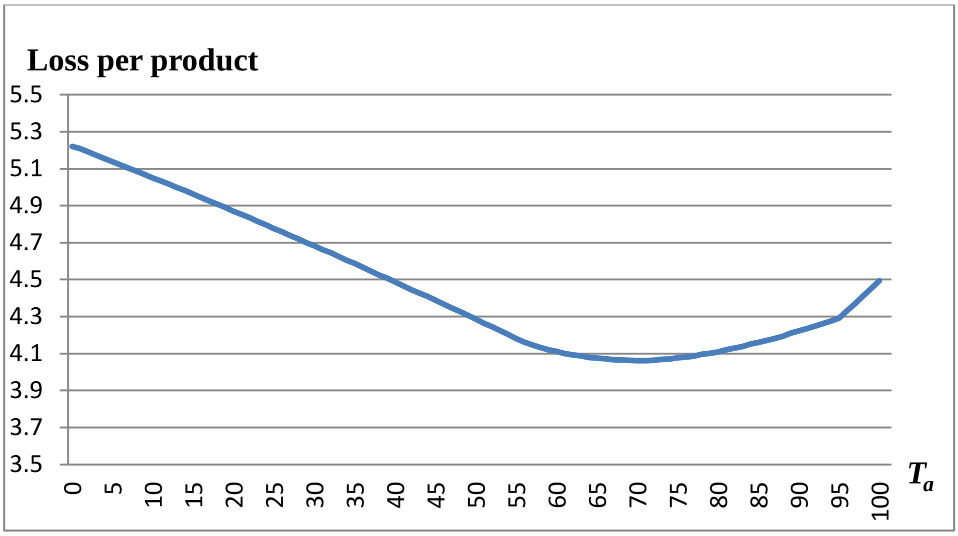



**Figure 5-3. Average loss per product against $T_a$ (STELLA)**

### 5.1.2 Simulation by MTB Program

With the MTB program developed in Chapter 4, it is not very demanding to check for the correctness of the proposed MTB model. After inputting the parameters, click the 'calculation' button, and the optimal results of both the “calculation” and “simulation” approaches are given automatically, and the loss per product curves are also shown in the left window, in which the black line stands for the simulation results and the red line for the calculation results. **Figure 5-4** shows the interface with the obtained results. It can be seen that the two “loss per product” curves are matched closely.

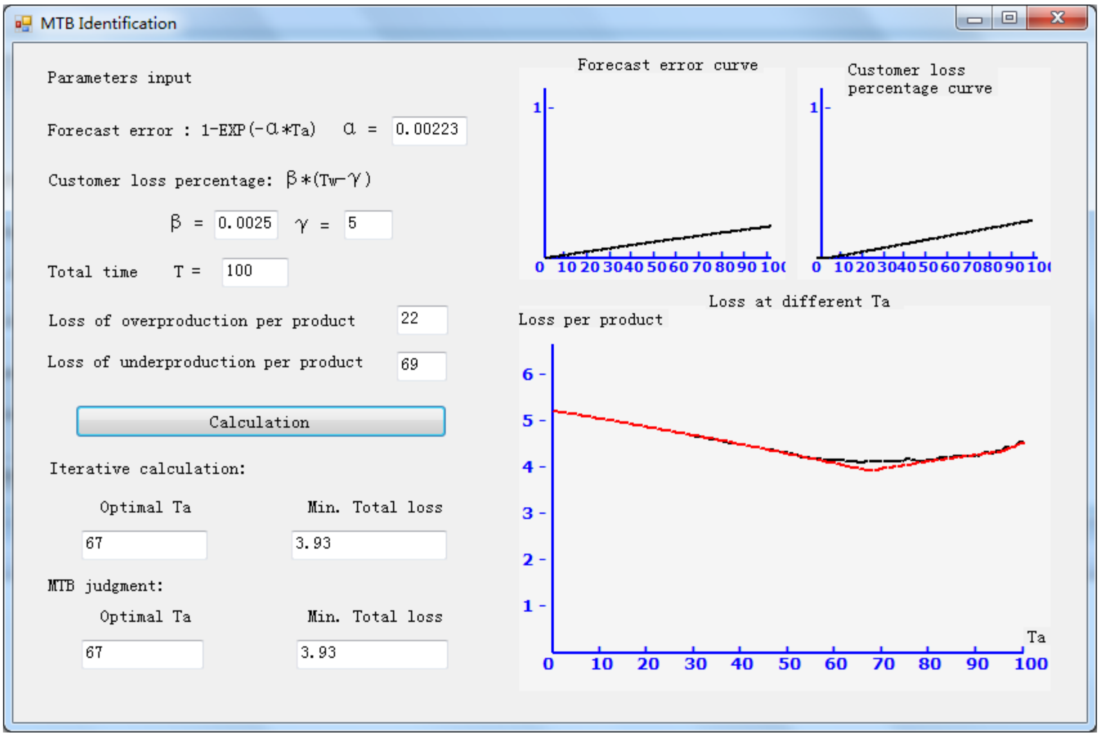



**Figure 5-4. MTB program interface**

### 5.1.3 MTB Optimal Judgment (Hand-Calculations)

Referring to the equations developed in **Chapter 3**,

$$T_{ax} = \frac{\alpha + \beta - \sqrt{(\alpha+\beta)^2 - 2\alpha^2\beta(T-\gamma)}}{\alpha^2} = 67.25 \approx 67$$

Therefore, the optimal production ahead time would be the one fulfilling the following conditions:

$$Min\ Z(T_a) = Min[Z(T_a = 0),\ Z(T_{ax} = 67),\ Z(T_a = 95)]$$

While $T_a = 0$, $T_w = T - T_a = 100$, one can find that $P_c(T_w) > P_f(T_a)$, then Equation [11] applies,

$$Z(T_a) = 0.25\left(1 - e^{-(\alpha \cdot T_a)}\right) \cdot L_u + \beta(T - T_a - \gamma) \cdot L_o$$

$$Z(T_a = 0) = 0.25\left(1 - e^{-(0.00223 \times 0)}\right) \cdot 69 + 0.0025(100 - 0 - 5) \cdot 22$$

$$\underline{\boldsymbol{Z(T_a = 0) = 5.225}}$$

When $T_{ax}$ = 67, $T_w$=$T$ - $T_a$ = 33, gives $P_c(T_w) < P_f(T_a)$, then **Equation [13]** should be used:

$$Z(T_a) = 0.5\left(0.5(1 - e^{-(\alpha \cdot T_a)}) + \beta(T - T_a - \gamma)\right) \cdot L_o + 0.5\left(1 - e^{-(\alpha \cdot T_a)}\right) \cdot L_u$$

$$Z(T_{ax} = 71) = 0.5\left(0.5 - 0.5e^{-(0.00223 \times 67)} + 0.0025 \cdot 28\right) \cdot 22 + 0.5\left(1 - e^{-(0.00223 \times 67)}\right) \cdot 69$$

$$\underline{\boldsymbol{Z(T_{ax} = 67) = 3.934}}$$

When $T_a$ = 95, $T_w$=$T$ - $T_a$ = 5, gives $P_c(T_w) < P_f(T_a)$,

$$Z(T_a) = 0.5\left(1 - e^{-(\alpha \cdot T_a)} + \beta(T - T_a - \gamma)\right) \cdot L_o + 0.5\left(1 - e^{-(\alpha \cdot T_a)}\right) \cdot L_u$$

$$Z(T_a = 95) = 0.5\left(1 - e^{-(0.00223 \times 95)}\right) \cdot 22 + 0.5\left(1 - e^{-(0.00223 \times 95)}\right) \cdot 69$$

$$\underline{\boldsymbol{Z(T_a = 95) = 4.343}}$$

As a result,

$$Min\, Z(T_a) = Min[Z(T_a = 0),\ Z(T_a = 67),\ Z(T_a = 95)]$$

$$\boldsymbol{Min\, Z(T_a) = Z(T_{ax} = 67)}$$

Hence, the optimal production ahead time $(T_a)$ is <u>67 time units,</u> that also is the interception point, and the profit loss is <u>3.934 dollars/unit</u>.

### 5.1.4 MTB Correctness Check

**Figure 5-5** shows the loss curve with $T_a$ ranging from [0, 100], in which the red line shows the STELLA simulation results, the blue line shows the MTB simulation results while the golden yellow one is the calculation result from MTB. To refresh the memory, the MTB simulation program used the random function to create the forecast error and it took an average value of the iteration results. In terms of the calculation approach in MTB, it used the average value of the even distribution range [0, $1 - e^{-(\alpha \cdot T_a)}$] to represent the forecast error.

It can be observed that the two simulation results and calculation result match well when $T_a$ increases, especially at the beginning stage where $T_a$ is small. This is because the forecast error is also small and so it has only a slight effect on production. The minimum point occurs approximately at [67, 3.9], which is quite similar in all three curves. When $T_a$ increases, the forecasting is more involved in the production. As the forecast error becomes larger, the differences in the three curves are expected and, normally, this situation can be improved by having more trial cycles in a simulation. The simulation and calculation results are also provided in **Appendix-C**.

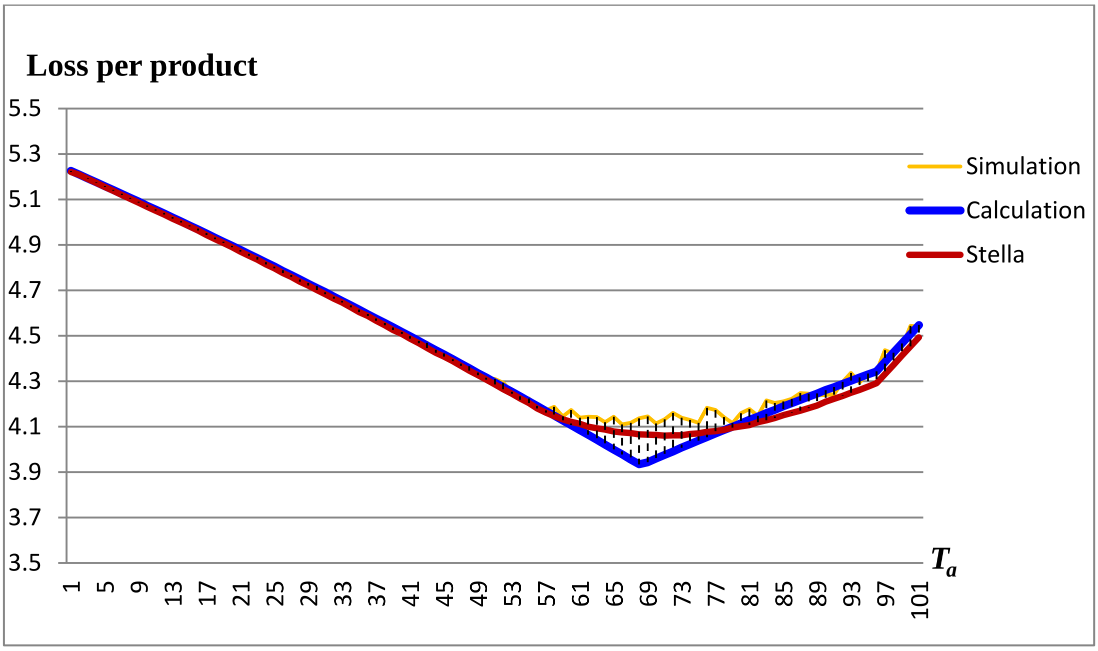



**Figure 5-5. Loss per product against $T_a$ [Simulation, Calculation and STELLA]**

## 5.2 Examination of MTB on Handling Extreme Situations

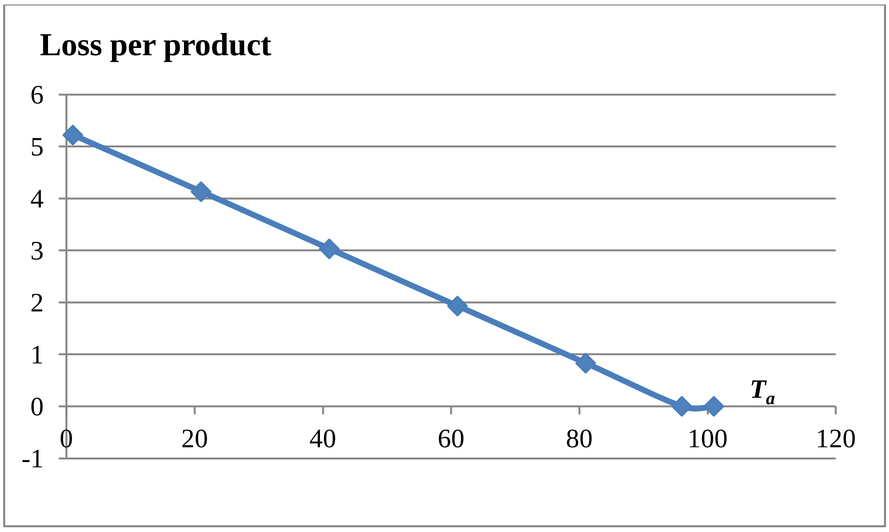
To validate the correctness of the MTB model, extreme situations were also examined. There are four extreme situations listed in **Table 5-2**. The following are the proofs of the model in handling these four extreme situations; other parameter settings can be referred back to **Table 5-1** given earlier.

**Table 5-2. Four extreme situations**

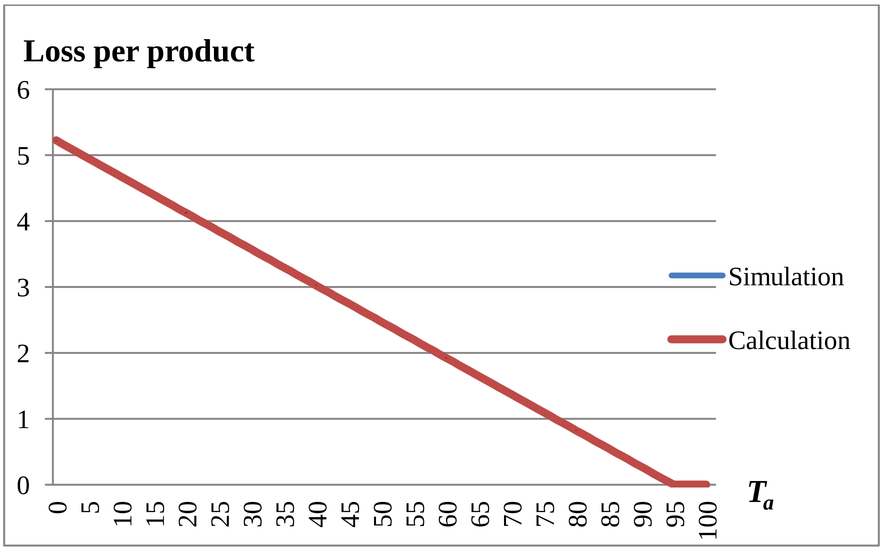
| Forecast Result | Customer Loyalty |
|---|---|
| 1. Totally accurate ($\alpha = 0$) | 3. Totally patience ($\beta = 0$ or $\gamma = T$) |
| 2. Totally inaccurate ($\alpha = \infty$) | 4. Totally impatience ($\beta = 1$ and $\gamma = 0$) |

1. In the totally forecast accurate ($\alpha = 0$) situation, since there is no forecast error, $P_f(T_a) = 1 - e^{-\alpha T_a} = 0$. Based on the optimal judgment conditions of the MTB model with $P_c(T_w) > P_f(T_a)$, $T_a = T - \gamma$ = 95 is the best result. The loss curves are plotted in **Figure 5-6** (a, b) where the minimum loss point is at (95, 0) and there is no loss at this point because the forecast is 100% accurate. It can be observed easily that the STELLA simulation, program simulation and calculation result match perfectly, and the customer loss in waiting in this circumstance has also been avoided because it is an ideal **MTS** case.

2. In the opposite situation of an extremely poor forecast [totally inaccurate forecast] ($\alpha = \infty$), the market demand forecast is completely useless. Subsequently, $T_a$ will not have any influence at all and $P_f(T_a) = 1$ always. In this circumstance, the minimum loss is $Z(T_a = 0) = 5.225$; also see Section 5.1.3 for more on the calculations. The curves are shown in **Figure 5-7**(a, b), where it can be observed easily that the STELLA simulation, calculation and program simulations results match well, and the minimum loss point is at about (0, 5.2), signifying that **MTO** should be a proper choice.

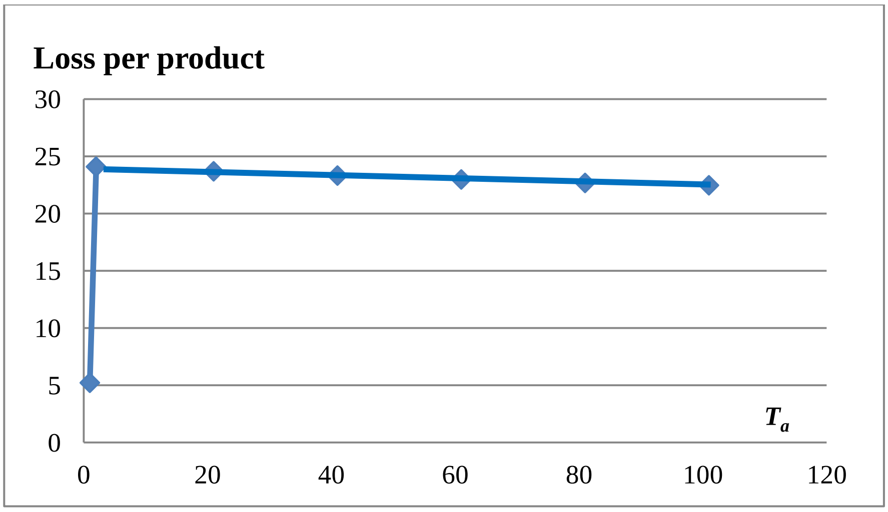



a: STELLA simulation

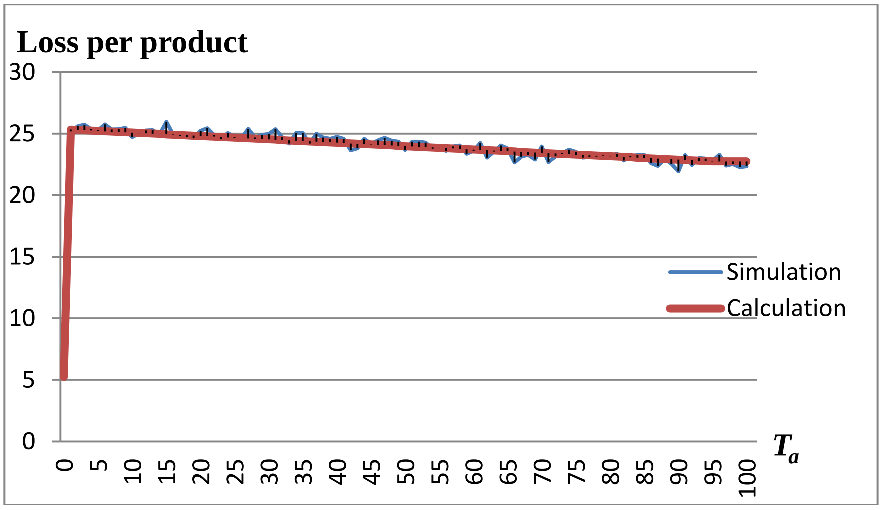



b: MTB program simulation and calculation

**Figure 5-6. Loss curves with $\alpha = 0$**



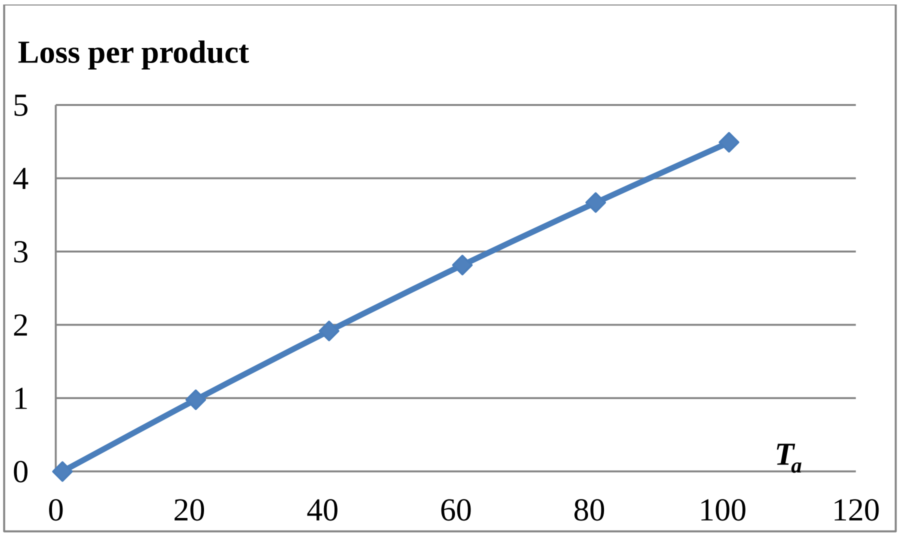

a: STELLA simulation



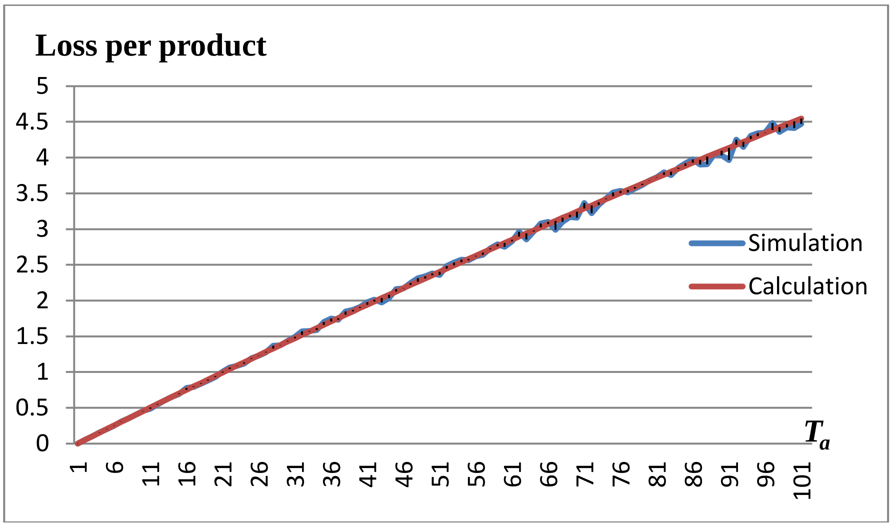

b: program simulation and calculation

**Figure 5-7. Loss curves with $\alpha = \infty$**

3. A totally patient customer ($\beta = 0$ or $\gamma = T$) means one who is either willing to wait indefinitely or at least to wait for the total production time. Thus, the customer loss percentage $P_c(T_w) = 0$. The curves are shown in **Figure 5-8** (a, b). One can see that the minimum loss point is at (0, 0) and this is just equivalent to the **MTO** as customers are willing to wait for the products.



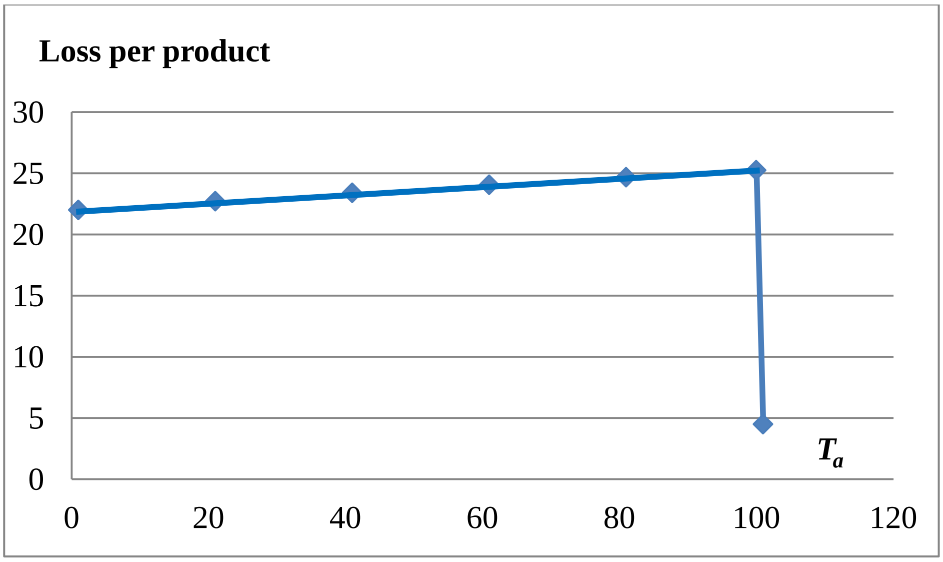

a: STELLA simulation



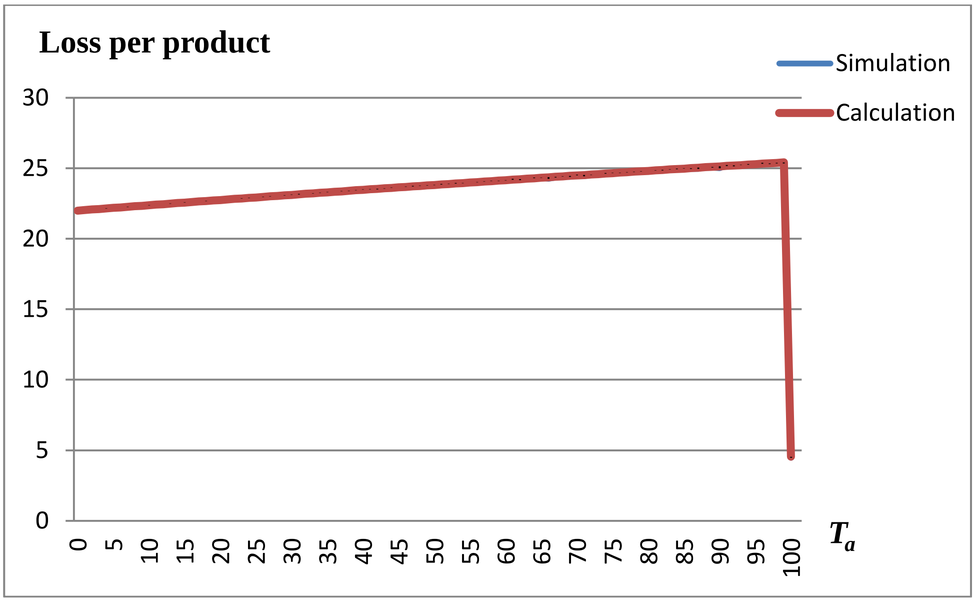

b: program simulation and calculation

**Figure 5-8. Loss curve with $\beta = 0$ and $\gamma = T$**

4. The last case is for totally impatient customers ($\beta = 1$ and $\gamma = 0$) and the loss curves are shown in **Figure 5-9**(a, b), where the minimum is at (100, 4.5). In this case, it is suggested to go for the **MTS** production approach as customers won't wait.




a: STELLA simulation




b: program simulation and calculation

**Figure 5-9. Loss curves with $\beta = 1$ and $\gamma = 0$**

Based on the results of the 5 cases presented (one general and four extreme cases), it can be seen that both the STELLA simulation and the MTB program simulation/calculation results match well with the outcomes of the proposed MTB model, as illustrated in **Section 5.1.3,** and, by means of hand calculations, the Smart MTB program works similarly. Thus, the unanimity of results not only confirms the correctness of the MTB model formulated in this research but also that the MTB simulation program code works well. The detailed results of the STELLA simulation, the MTB simulation and the calculation of these four extreme cases are shown in **Appendix-D**. Since the MTB simulation is able to produce results and curves automatically, with little post processing work, the rest of the tests used the MTB simulation as a tool to examine the effects of various parameters involved in the model.

## 5.3 MTB Parameters Analysis

In fact, the forecast errors and the customer behavior in regard to different products are diverse, so it is interesting to understand the implications of the associated parameters on the MTB model. The following investigations concentrate on the three main parameters in the MTB, $\beta, \gamma$ and $\alpha$, such that, while examining one parameter, all others are kept constant.

### Implication of Forecast Error Parameter ($\alpha$) on MTB

Referring to the forecast error model employed in this research (see **Section 3.2.1**), the parameter $\alpha$ plays an important role because the larger the value of $\alpha$, the bigger the forecast error. **Figure 5-10** shows the loss curves with $\alpha$ changing from 0.004 to 0.001 in steps of 0.0005. Indeed, this is equivalent to the forecast error dropping from 33% to 9.5%. **Table 5-3** summaries the results with ($\beta = 0.0025$ and $\gamma = 5$), and letting $T_{ax}$ be the specific $T_a$ at the turning point on each plotted line. Related data can be obtained in **Appendix-E**.

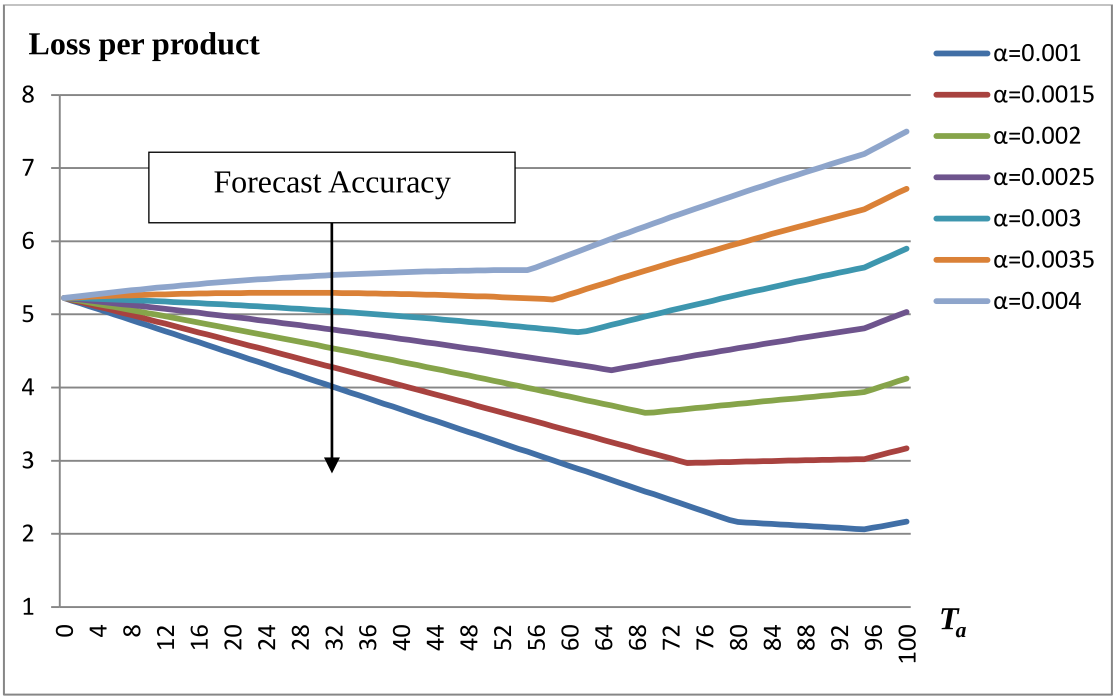



**Figure 5-10. Loss curves at different $\alpha$**

**Table 5-3. Loss at different α**

| $\alpha$ | $T_{ax}$ | Min[Z(0), Z($T_{ax}$), Z(95)] | Min Z($T_a$) | Decision |
|---|---|---|---|---|
| 0.0040 | 55 | Z(0) | (0, 5.225) | MTO |
| 0.0035 | 58 | Z(0) | (0, 5.225) | MTO |
| 0.0030 | 61 | Z(61) | (61, 4.755) | MTB |
| 0.0025 | 65 | Z(65) | (65, 4.237) | MTB |
| 0.0020 | 69 | Z(69) | (69, 3.654) | MTB |
| 0.0015 | 74 | Z(74) | (74, 2.968) | MTB |
| 0.0010 | 80 | Z(95) | (95, 2.062) | MTS |

The x-axis in **Figure 5-10** is the parameter $T_a$, from 0 to 100, which covers the total production time ($T$) required. In reality, ($T_a = 0$) means the MTO production and ($T_a \geq T - \gamma$) is MTS. Referring to the results, MTB is appropriate for $\alpha$ values from 0.0015 to 0.0030. It is observed that, with the value of $\alpha$ dropping from 0.0040 to 0.0010, the loss becomes lower; this is simply because the loss decreases as the forecast error becomes smaller. Moreover, when it is so small ($\alpha \leq 0.001$), it tends to favor MTS as the demand forecast is quite reliable in such a case. Another interesting point is that all curves start at the same point (0, 5.225) where the forecast error does not influence the production because it is an MTO case. A general picture is that when $\alpha$ decreases, the optimal production approach shifts from MTO towards MTS.

**Implication of Customer Loss in Waiting Parameters ($\beta$ and $\gamma$) on MTB**

Referring to the modelling of the customer loss in waiting in Section 3.2.2, $\beta$ and $\gamma$ represent the customer behavior in waiting, such that a smaller $\beta$ value stands for better customer loyalty against time, while $\gamma$ specifies the duration for which all customers are willing to wait. First, we examined the change by keeping $\alpha$ and $\gamma$ constant; one can also refer back to **Table 5-1** for the parameter settings. **Figure 5-11** shows the plots of $\beta$ from 0.001 to 0.004 in 0.0005 increments, which means the customer loss at the end of the waiting time $T$ (=100) is in the range of 10% to 40%, where an increment of 5% each step was applied to examine the outcomes in this range. **Table 5-4** abstracts the loss at different values of $\beta$ and the detailed data can be seen in **Appendix-F**.

**Table 5-4. Loss at different $\beta$**

| $\beta$ | $T_{ax}$ | Min[Z(0), Z($T_{ax}$), Z(95)] | Min Z($T_a$) | Decision |
|---|---|---|---|---|
| 0.0010 | 46 | Z(0) | (0, 2.090) | MTO |
| 0.0015 | 56 | Z(0) | (0, 3.135) | MTO |
| 0.0020 | 62 | Z(62) | (62, 3.679) | MTB |
| 0.0025 | 67 | Z(67) | (67, 3.934) | MTB |
| 0.0030 | 71 | Z(71) | (71, 4.123) | MTB |
| 0.0035 | 74 | Z(74) | (74, 4.269) | MTB |
| 0.0040 | 76 | Z(76) | (95, 4.343) | MTS |

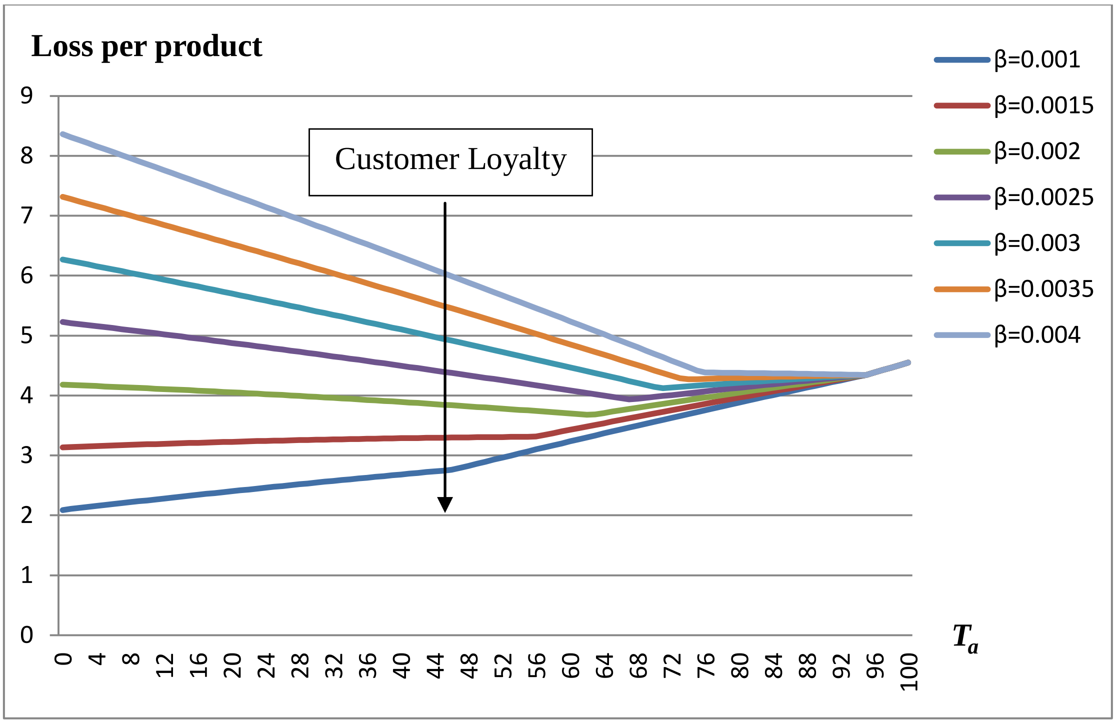



**Figure 5-11. Loss curves at different $\beta$**

When $\beta$ increases, the turning point $T_{ax}$ increases as well because products have to be made in advance to reduce the waiting time, since there are many impatient customers! Furthermore, all curves end at the same point (95, 4.547), indicating that customer loss due to waiting has no effect if it is a MTS production, simply because there is no more waiting.

$\gamma$ is the time that all customers are willing to wait and **Figure 5-12** shows the loss with different $\gamma$ values from 5 to 35, in steps of 10. Similarly, **Table 5-5** gives a brief summary, and more information is given in **Appendix-G**. It is easy to observe that the loss reduces while the $\boldsymbol{\gamma}$ value increases, which means the customers' patience. In addition, $T_{ax}$ shifts towards the left as the $\boldsymbol{\gamma}$ value increases.

**Table 5-5. Loss at different $\gamma$**

| $\gamma$ | $T_{ax}$ | $Min[Z(0), Z(T_{ax}), Z(95)]$ | $Min\ Z(T_a)$ | Decision |
|---|---|---|---|---|
| 5 | 67 | Z(67) | (67, 3.934) | MTB |
| 15 | 60 | Z(60) | (60, 3.537) | MTB |
| 25 | 53 | Z(53) | (53, 3.141) | MTB |
| 35 | 46 | Z(46) | (46, 2.74) | MTB |

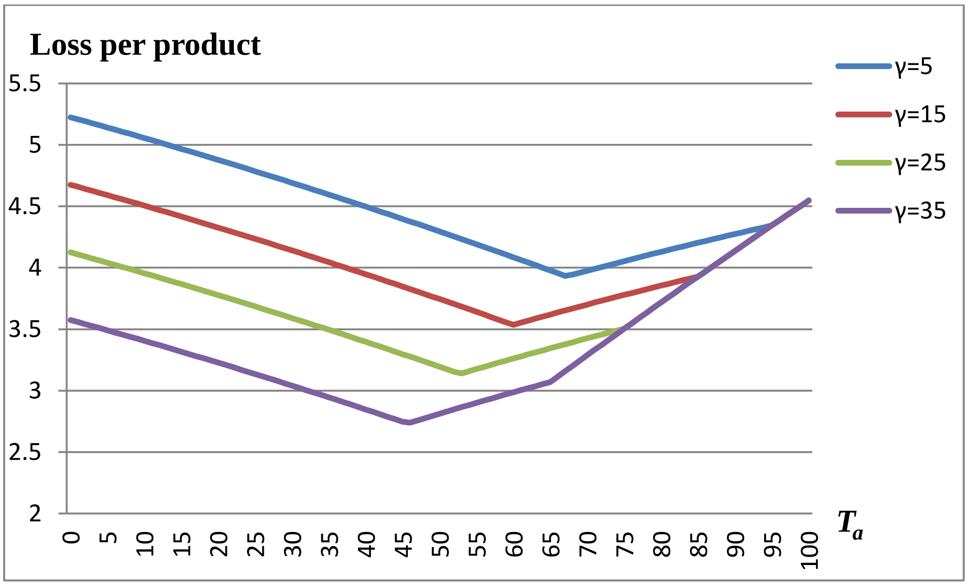



**Figure 5-12. Loss curves at different $\gamma$**

In conclusion, while the value of $\alpha$ increases, there is a larger forecast error and the optimal production approach is favored more towards MTO. When $\boldsymbol{\beta}$ decreases or the value of $\boldsymbol{\gamma}$ increases, customers become more loyal so the optimal production approach shifts to MTO.

## 5.4 Comparison of MTB Judgment Equations [w/o 2rd order term]

Now, recalling the two $T_a$ calculation **Equations [18]** and **[19]** formulated in chapter 3 for the interception point ($T_{ax}$) determination:

$$T_{ax} = \frac{\alpha+\beta-\sqrt{(\alpha+\beta)^2-2\alpha^2\beta(T-\gamma)}}{\alpha^2} \qquad [18]$$

$$T_{ax} = \frac{\beta(T-\gamma)}{(\alpha+\beta)} \qquad [19]$$

In comparison with the Iterative calculation, **Table 5-6** to **Table 5-8** show the accuracies of these two equations with the same system configurations as before (results are rounded off to the nearest integer).

**Table 5-6. $T_{ax}$ accuracy at different $\alpha$**

| $T$=100, $\beta$=0.0025 $\gamma$=5 | Iterative calculation $T_{ax}$ | Equation [18] | | Equation [19] | |
|---|---|---|---|---|---|
| | | $T_{ax}$ | Error | $T_{ax}$ | Error |
| $\alpha$=0.0030 | 62 | 61 | -1.6% | 59 | -4.8% |
| $\alpha$=0.0025 | 66 | 65 | -1.5% | 63 | -4.5% |
| $\alpha$=0.0020 | 69 | 69 | 0 | 68 | -1.4% |
| $\alpha$=0.0015 | 71 | 74 | 4.2% | 73 | 2.8% |

**Table 5-7. $T_{ax}$ accuracy at different $\beta$**

| $T$=100, $\alpha$=0.0025 $\gamma$=5 | Iterative calculation $T_{ax}$ | Equation [18] | | Equation [19] | |
|---|---|---|---|---|---|
| | | $T_{ax}$ | Error | $T_{ax}$ | Error |
| $\beta$=0.0020 | 62 | 62 | 0 | 58 | -6.5% |
| $\beta$=0.0025 | 66 | 67 | 1.5% | 63 | -4.5% |
| $\beta$=0.0030 | 71 | 71 | 0 | 67 | -5.6% |
| $\beta$=0.0035 | 76 | 74 | -2.6% | 70 | 7.9% |

The results show that **Equation [18]** gives higher accuracies in all cases, except at $\alpha$=0.0015, as shown in **Table 5-6**. However, **Equation [19]** is simpler and the accuracies were still better than 92%. In fact, by taking into consideration the complications in real-life situations, it is quite adequate. However, with the computational help, calculations are not a burden anymore and the choice of either equation is a matter of personal choice.

**Table 5-8. $T_{ax}$ accuracy at different $\gamma$**

| $T$=100, $\alpha$=0.0025 $\beta$=0.0025 | Iterative calculation $T_{ax}$ | Equation [18] | | Equation [19] | |
|---|---|---|---|---|---|
| | | $T_{ax}$ | Error | $T_{ax}$ | Error |
| $\gamma$=5 | 66 | 67 | 1.5% | 63 | -4.5% |
| $\gamma$=15 | 59 | 60 | 1.7% | 57 | -3.4% |
| $\gamma$=25 | 54 | 53 | -1.9% | 50 | -7.4% |
| $\gamma$=35 | 45 | 46 | -2.2% | 43 | -4.4% |

## 5.5 MTO/MTB/MTS Performance Analysis

With the help of the MTB model, it would be sensible to study the effect of selecting a different production approach (MTO, MTB or MTS) for a production system. **Table 5-9** to **Table 5-11** are the results of applying various approaches. Actually, there is no particular approach that can always be considered to be the best. However, it can also be observed that the result can have quite a big difference. For example, in **Table 5-9** with $\alpha$=0.0015, MTB gave the lowest cost loss. In comparison, MTO would have about 76% more loss per product ($\frac{5.225-2.968}{2.968} = 76\%$) than MTB and would certainly not be a good choice.

**Table 5-9. MTO/MTB/MTS at different $\alpha$**

| $T$=100, $\beta$=0.0025 $\gamma$=5, $L_u$=69, $L_o$=22 | Loss per product | | |
|---|---|---|---|
| | MTO | MTB | MTS |
| $\alpha$=0.0040 | 5.225 | 5.606 | 7.50 |
| $\alpha$=0.0035 | 5.225 | 5.204 | 6.718 |
| $\alpha$=0.0030 | 5.225 | 4.755 | 5.896 |
| $\alpha$=0.0025 | 5.225 | 4.237 | 5.032 |
| $\alpha$=0.0020 | 5.225 | 3.654 | 4.124 |
| $\alpha$=0.0015 | 5.225 | 2.968 | 3.169 |
| $\alpha$=0.0010 | 5.225 | 2.190 | 2.165 |

**Table 5-10. MTO/MTB/MTS at different *β***

| $T$=100, α=0.0025 $\gamma$=5, $L_u$=69, $L_o$=22 | Loss per product | | |
|---|---|---|---|
| | MTO | MTB | MTS |
| $\beta$=0.0010 | 2.09 | 2.76 | 4.547 |
| $\beta$=0.0015 | 3.135 | 3.314 | 4.547 |
| $\beta$=0.0020 | 4.18 | 3.679 | 4.547 |
| $\beta$=0.0025 | 5.225 | 3.934 | 4.547 |
| $\beta$=0.0030 | 6.27 | 4.123 | 4.547 |
| $\beta$=0.0035 | 7.315 | 4.269 | 4.547 |
| $\beta$=0.0040 | 8.36 | 8.22 | 4.547 |

**Table 5-11. MTO/MTB/MTS at different *γ***

| $T$=100, α=0.0025 $\beta$=0.006, $L_u$=59, $L_o$=14 | Loss per product | | |
|---|---|---|---|
| | MTO | MTB | MTS |
| $\gamma$=5 | 5.225 | 3.934 | 4.547 |
| $\gamma$=15 | 4.675 | **3.537** | 4.547 |
| $\gamma$=25 | 4.125 | **3.141** | 4.547 |
| $\gamma$=35 | 3.575 | **2.74** | 4.547 |

It can be found in the above three tables that both MTS and MTO can achieve the best performance in some situations; however, there are still some situations in which neither the MTS nor the MTO approaches perform best, and the MTB approach fills the gap. This indicates clearly that, in the industrial production, MTB is a valuable approach which can provide better production performance than MTS or MTO. Therefore, in the case when both the MTO and the MTS approaches do not well fit, MTB will be a wise option.

## 5.6 Illustrative Examples

After a full examination of the MTB model, it is the time to illustrate how to use the developed model in a practical situation. Here, two examples are shown below.

**Case One**:

*A bicycle manufacturer supplies road bikes to retailers and it has been estimated that the profit is $200 per bicycle. The production time (including delivery time) of a bicycle is 32 days. Marketing research showed that customers normally change their minds within one day, and there is a 1% customer loss with every single day passed. The average loss of fail-to-sale is $35 per bicycle. The demand forecast shows that 1500 bikes could be sold in 32 days and the error is 15%.*

First, we summarize the related factors as follows:

At $T_a = T,\ P_f(T_a) = 15\%$. I.e.,

$$0.15 = 1 - e^{-(32\alpha)}$$

$$\alpha = \frac{\ln 0.85}{-32} = 0.005078 \cong 0.0051$$

Therefore, the values of parameters are found as listed in **Table 5-12**.

**Table 5-12. The value of parameters**

| $\boldsymbol{T}$ | $\boldsymbol{\alpha}$ | $\boldsymbol{\beta}$ | $\boldsymbol{\gamma}$ | $\boldsymbol{L_u}$ | $\boldsymbol{L_o}$ | $\boldsymbol{X}$ |
|---|---|---|---|---|---|---|
| 32 | 0.0051 | 0.01 | 1 | 200 | 45 | 1500 |

$T_a$ at the interception point can be determined by:

$$T_{ax} = \frac{\alpha + \beta - \sqrt{(\alpha+\beta)^2 - 2\alpha^2\beta(T-\gamma)}}{\alpha^2} = 25.03 \cong 25$$

Now, recall the objective function:

$$Min\ Z(T_a) = Min[Z(T_a = 0),\ Z(T_a = T_{ax}),\ Z(T_a = 31)]$$

While $T_a = 0$ [MTO], $T_w = T - T_a = 32$, so **Equation [11]** applies,

$$Z(T_a) = 0.25\left(1 - e^{-(\alpha \cdot T_a)}\right) \cdot L_u + \beta(T - T_a - \gamma) \cdot L_o$$

$$Z(T_a = 0) = 0.25\left(1 - e^{-(0.0033 \times 0)}\right) \cdot 200 + 0.012(32 - 0 - 1) \cdot 35$$

$\underline{\boldsymbol{Z(T_a = 0) = 13.95}}$ **[MTO]**

While $T_a$ =25 [MTB], $T_w$ =$T$ - $T_a$ =7, we have $P_c(T_w) < P_f(T_a)$, and **Equation [13]** should be used:

$$Z(T_a = 25) = 0.5\left(0.5(1 - e^{-(0.0033 \cdot 25)}) + 0.012(32 - 25 - 1)\right) \cdot 200$$

$$+0.5(1 - e^{-(0.0033 \cdot 25)}) \cdot 35$$

$\underline{\boldsymbol{Z(T_a = 25) = 8.66}}$ **[MTB]**

While $T_a$ = 31 [MTS], $T_w$ =$T$ - $T_a$ = 1, **Equation [13]** is used:

$$Z(T_a = 31) = 0.5\left(0.5(1 - e^{-(0.0033 \cdot 31)}) + 0.012(32 - 31 - 1)\right) \cdot 200$$

$$+ 0.25\left(1 - e^{-(0.0033 \cdot 31)}\right) \cdot 35$$

$\underline{\boldsymbol{Z(T_a = 31) = 8.92}}$ **[MTS]**

Now, recall the objective function as:

$$Min\, Z(T_a) = Min[Z(T_a = 0),\ Z(T_a = 25),\ Z(T_a = 31)]$$

$$\boldsymbol{Min\, Z(T_a) = Z(T_a = 25)}$$

In the simulation, the lowest loss point is at 26 time units and the loss is 8.88 per product. **Table 5-13** provides a comparison with the iterative calculation and the MTB judgment results. It can be observed that the errors between these two results, both $T_a$ and loss, are quite small and acceptable. **Figure 5-13** shows the curves of loss per product against $T_a$.

**Table 5-13. Optimal result comparison**

| | Iterative Calculation | MTB Judgment | Deviation |
|---|---|---|---|
| $T_a$ | 25 | 25 | 0% |
| Loss at $T_a$ | 8.69 | 8.66 | -0.3% |

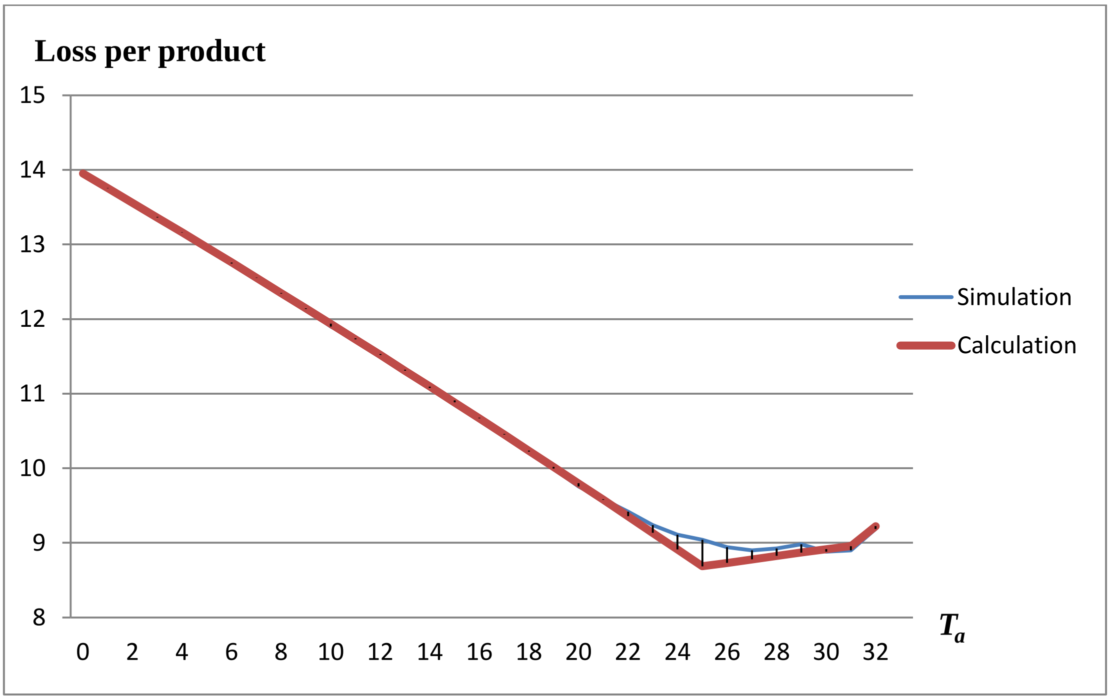



**Figure 5-13. Loss per product at different $T_a$**

So the final solution is that we should have a production ahead time of 25 days and the loss is about \$8.66/unit. Therefore, the expected total loss of this batch of 1500 bikes, by taking into the consideration forecast error and customer loss in waiting, is:

1500 x \$8.66 = \$ 12,990

Hence, the expected total profit is:

1500 x (\$200-\$8.66) = \$ 287,010

Indeed, we can also make use of the developed Smart MTB program to assist the calculations, and see the outcomes in **Figure 5-14** are just the same.

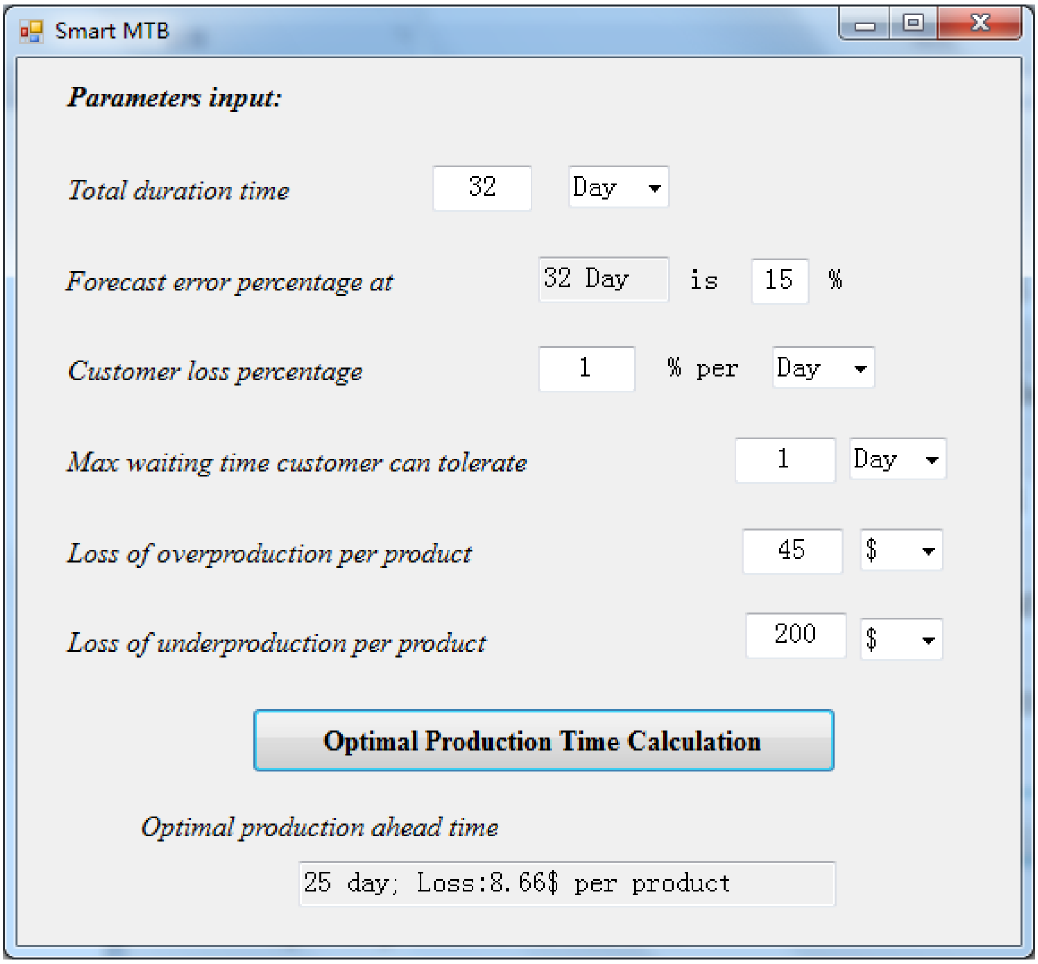



**Figure 5-14. Smart MTB Interface - Illustration Example**

**Case Two**:

Besides the manufacturing field, the applications of MTB can also be extended to the logistics area, such as the order management in a supermarket. For example, one brand of television is selling at $1699, and the delivery time from the distributor is 8 days. The gross profit is 15% of the selling price ($1699x0.15≈$255), and marketing research showed that customers are willing to wait for 2 days but that 20% of customers will opt for other alternatives afterwards. The average loss of fail-to-sale has been estimated to be $56 per TV. The demand forecast error is 35% with a time span of 8 days. Then, with the help of the Smart MTB program developed, we obtained the results shown in **Figure 5-15**.

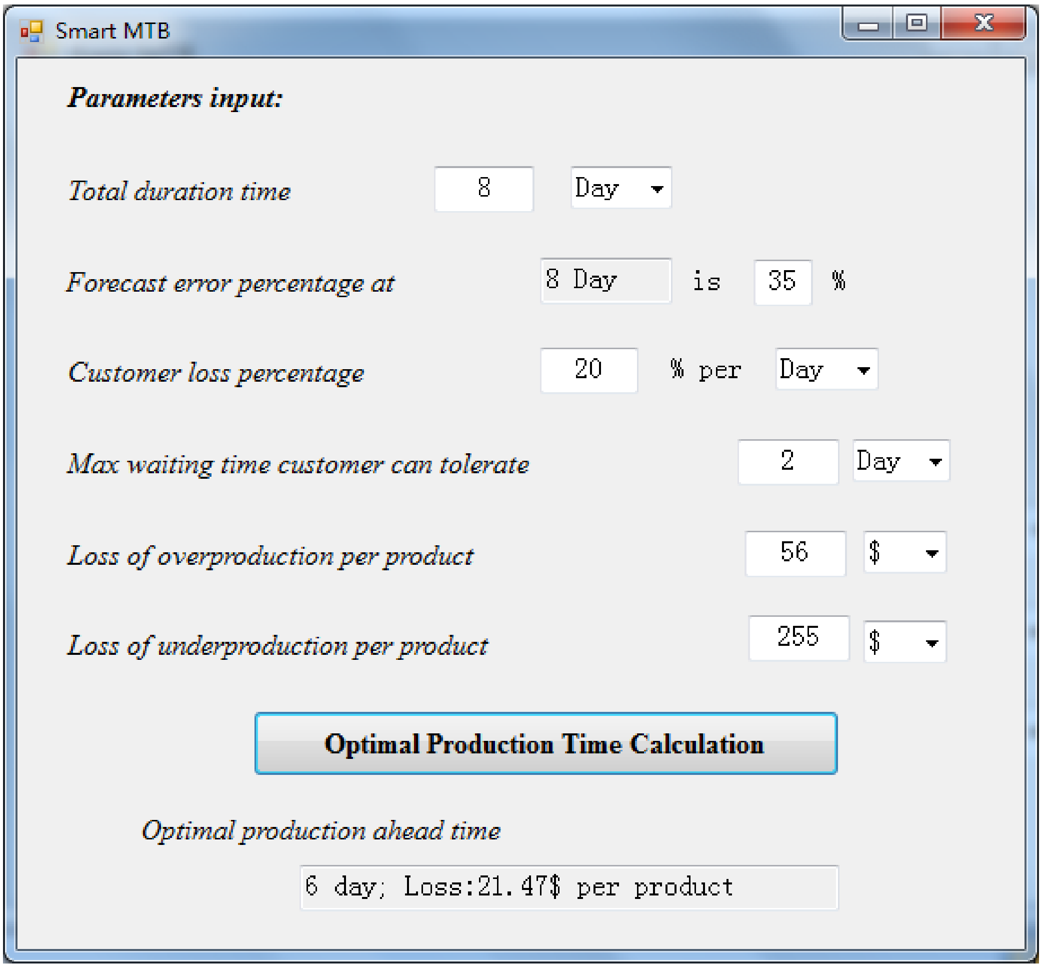



**Figure 5-15. Smart MTB – Logistics Example**

We can see that it will be better to place an order 6 days in advance so that the possible loss would be minimal (6 days, $21.47 per product); in this case, we simply considered the production ahead time as the advance time to place an order. In general, the application of MTB can be extended to situations with uncertain demand, including manufacturing, sales, logistics, transportation etc.

# Chapter 6 Discussion and Conclusion

The proposed MTB has been demonstrated to be useful in determining a suitable production policy. In addition, it also has potential for being used in other kinds of systems. In this section, the contribution and the findings of this research project are first presented, some possible applications of MTB are then explored and, finally, some prospective directions for further related research are discussed.

## 6.1 Research Findings and Conclusions

The prime objective of this research was to put forward a new cost effective production policy that is able to handle demand uncertainty by taking into the consideration customers' loyalty in terms of delivering efficiency, and the major contribution is the formulated novel MTB (Make-To-Balance) production decision method. Through this research, the concepts of MTB production and its coupled algorithms were generated. To ease the calculation work involved, the Smart MTB program was also coded by using Microsoft Visual C#. The strength of the MTB model is that it tries to compensate the demand forecast error and the customer loss in waiting on a production system by providing an appropriate production start "ahead" time. In comparison with the two well-known MTO and MTS production approaches, the suggestion of a suitable production start ahead time is an important input to the production arrangement. In reality, MTB can enable production to react to the changing market more accurately because it facilitates the use of the demand forecast in a smarter manner, and this definitely helps in production management.

This research has shown that MTB can be regarded as a general production style, while MTO and MTS are the two boundary cases because an optimal solution of the MTB model can also turn out to be either MTO or MTS on severe occasions. To verify the success of MTB, different situations have been examined, including some extreme cases, and the results proved that the MTB model works well in all cases.

In the marketplace, more choices with shorter delivery times are always preferred by customers, and the way in which a manufacturer fulfills the customers' requirements in a cost effective manner is a key to success. It is not difficult to see that MTB helps management to decide on an appropriate time for a batch of products to start production. Moreover, it is not only tailor made for a particular production system and it does not require any physical change in the original system, such as redesigning the workflow, facility layout, etc. to make it work.

Forecasting always plays an important role in production management. This research has examined the related factors, such as how the forecast error interacts with the customer loss in waiting, and the MTB fills the gap with regard to managing the forecast error and the customer loss in waiting. In the mathematical model, $\alpha$ represents the forecast error level such that, when the value of $\alpha$ increases by 0.005, the forecast error level rises by about 4%, as described in **Chapter 5,** and the total loss is quite notable in each small 0.005 step. This consequence shows clearly that, while the forecast accuracy increases a little ($\alpha$ decreases), a great improvement can be obtained. It also signifies that, when the forecast accuracy improves, the total production loss decreases and the optimal manufacture approach shifts towards the MTS side.

Additionally, this research has extended the previous production studies by incorporating customer behaviour, as it affects the production arrangements considerably. It also indicates that, with a more loyal (or patient) customer base, a lower production loss can be achieved and the optimal solution is biased to the MTO side, in general. Therefore, proper marketing strategies, such as building a strong brand name, providing quality products, and having a proper complaint handling system, can be well worth the investment. On the other hand, when there are impatient customers, the production loss can be high due to waiting, and the optimal solution goes towards the MTS.

There are two more factors affecting the optimal MTB solution, which are overproduction and underproduction losses. The results show that, with a higher overproduction loss (or the lower underproduction loss), the optimal solution shifts to the MTO side. On the other hand, the higher underproduction loss (or the lower overproduction loss) makes it favor the MTS.

## 6.2 Further Developments of MTB

This section discusses some new directions for further research based on the MTB. In the current MTB model, there are five key elements involved: total time ($T$), overproduction loss ($L_o$), underproduction loss ($L_u$), forecast error ($P_f(T_a)$) and customer loss in waiting ($P_c(T_w)$). In order to shorten the calculation, the overproduction and underproduction losses per product are considered constants. However, in real life, the manufacturing cost is changing with time; for example, the raw material price varies with time, there could be quantity discounts, the inventory cost increases with quantity and storage time, etc. Therefore, it will be beneficial if these sorts of variations can be reflected in the overproduction and underproduction loss functions in the MTB model. It can also be seen from the research findings that the values of overproduction and underproduction losses will steer a MTB solution towards either MTO or MTS. Therefore, the development of a threshold ratio of

"$L_o/L_u$" may benefit decision making.

We have adopted an exponential distribution function to represent the forecast error and a piecewise linear function to model the customer loss in waiting. They assist the explanation of how the MTB works in this research, but other possible functions imitating the forecast error and the customer loss in waiting should also be investigated in the future in order to cater for different scenarios. In addition, customer behavior can be affected by the brand name effect, product nature, price, service level, age group, gender, etc., therefore there is still a lot of work to be done to make the model more extensively in the future.

Furthermore, multi-product manufacturing is also an important sector in production and this leads to mix-products on the production floor and their throughput times, profits, loss due to shortages, etc. are different, so tackling multi-products production is another challenging area to be investigated. To determine the optimal production ahead time based on the given forecast error and the customer loss information is the main purpose of the MTB. In fact, it would be helpful if it could also suggest the optimal production quantity in association with the given production ahead time. Finally, MTO and MTS are widely adopted production approaches and there have been some studies on hybrid MTO/MTS in order to address customers' needs in a better way. Thus, MTB is a research direction to be more explored deeply.

## 6.3 Summary of Research Achievement and Contribution

In conclude, this research developed a production control model called MTB that can be viewed as way to determine the most suitable production starting time, and the solution always lies between the MTO and MTS time horizon such that

MTO and MTS are considered as the boundary cases. The basic idea is to strike a balance in customer waiting and forecast accuracy. The thesis provides the rationale of the MTB model along with a complete numerical illustrations. The problem itself is not trivial and can be implemented in reality, especially considering the proposed model a starting point of some more related researches in the future. Lastly, this research also generates useful managerial insights for production management.

## References


Aghezzaf, E.-H., Sitompul, C., & Najid, N. M. (2010). Models for robust tactical planning in multi-stage production systems with uncertain demands. *Computers & Operations Research, 37*, pp. 880-889.

Aigner, D., & Goldfeld, S. (1974). Estimation and prediction from aggregate data when aggregates are measured more accurately than their components. *Econometrica, 42*, pp. 113–134.

Alfares, H. K., & Nazeeruddin, M. (2002). Electric load forecasting: literature survey and classification of methods. *International Journal of Systems Science, 33*, pp. 23-34.

Aliev, R., Fazlollahi, B., Guirimov, B., & Aliev, R. (2007). Fuzzy-genetic approach to aggregate production – distribution planning in supply chain management. *Information Sciences, 177*, pp. 4241–4255.

Anderson, E., Fitzsimons, G., & Simester, D. (2006). Mitigating the cost of stockouts. *Management Science, 52*, pp. 1751–1763.

Andreassen, W. T., & Lindestad, B. (1998). Customer loyalty and complex services. *International Journal of Service Industry Management, 9*, pp. 7-23.

Aysegul, S., Nabil, A., & Stéphane, D.-P. (2010). A literature review on the impact of RFID technologies on supply chain management. *International Journal of Production Economics, 128*, pp. 77-95.

Bae, J., Kim, S., & Lee, E. Y. (2001). The Virtual Waiting Time of the M/G/1 Queue with Impatient Customers. *Queueing Systems, 38*, pp. 485-494.

Bardacki, A., & Whitelock, J. (2004). How "ready" are customers for mass customization? An exploratory investigation. *European Journal of Marketing, 38*, pp. 396–416.

Benjaafar, S., ElHafsi, M., & Huang, T. (2010). Optimal control of a production-inventory system with both backorders and lost sales. *Naval Research Logistics, 57*, pp. 252–265.

Bert, v. W. (2007). Large infrastructure projects: a review of the quality of demand forecasts and cost estimations. *Environment and Planning B: Planning and Design, 34*, pp. 611-625.

Bertrand, J., & Rutten, W. (1999). Evaluation of three production planning procedures for the use of recipe flexibility. *European Journal of Operational Research, 115*, pp. 179-194.

Bonvik, A., Couch, C., & Gershwin, S. (1997). A comparison of production-line control mechanisms. *International Journal of Production Research, 25*, pp. 789–804.

Botta, V., Guinet, A., & Boulle, D. (1997). Object-oriented analysis with structured and integrated specifications and solutions (OASISS) for production system control. *Journal of Intelligent Manufacturing, 8*, pp. 3-14.

Boylan, J. E., & Syntetos, A. A. (2010). Spare parts management: a review of forecasting research and extensions. *IMA Journal of Management Mathematics*, *21*, pp. 227−237.

Brabazona, P. G., & MacCarthy, B. (2012). Investigating along tail in retail vehicle sales. *Omega, 40*, pp. 302-313.

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Zeltyn, S., Zhao, L., & Haipeng, S. (2005). Statistical analysis of a telephone call center:a queueing-science perspective. *Journal of the American Statistical Association, 100*, pp. 36–50.

Brynjolfsson, E., Hu, Y., & Simester, D. (2011). Goodbye Pareto Principle, Hello Long Tail: The Effect of Search Costs on the Concentration of Product Sales. *Management Science, 57*, pp. 1373-1386.

Burbidge, J. (1971). *The Principles of Production Control* (third edition ed.). London:MacDonald & Evans.

Byrne, M., & Hossain, M. (2005). Production planning: An improved hybrid approach. *International Journal of Production Economics, 93-94*, pp. 225-229.

Carrie, A. (1975). The layout of multi-product lines. *International Journal of Production Research, 13*, pp. 541-557.

Chan, C., & Tang, C. (2007). Formulation of a stock minimisation algorithm for the management of inbound logistics. *Int. J. Enterprise Network Management, 1*, pp. 394-405.

Chang, S.-H., Pai, P.-F., Yuan, K.-J., Wang, B.-C., & Li, R.-K. (2003). Heuristic PAC model for a hybrid MTO and MTS production environment. *International Journal of Production Economics, 85(3)*, pp. 347-358.

Chen, A., & Blue, J. (2010). Performance analysis of demand planning approaches for aggregating, forecasting and disaggregating interrelated demands. *International Journal of Production Economics, 128*, pp. 586–602.

Chen, H. (., Sharon, N., & Akshay, R. R. (2005). Cultural Differences in Consumer Impatience. *Journal of Marketing Research*, *XLII*, pp. 291-301.

Chiang, D., Guo, R., & Pai, F. (2008). Improved customer satisfaction with a hybrid dispatching rule in semiconductor back-end factories. *International Journal of Production Research, 46(17)*, pp. 4903–4923.

Choobineh, F., & Sowrirajan, S. (1996). *Capacitated- Constant Work in Process (C-CONWIP): a Job Shop Control System.* University of Nebraska, Industrial and Management Systems Engineering, Lincoln, Nebraska.

Chung, J., & Jang, J. (2009). A WIP Balancing Procedure for Throughput Maximization in Semiconductor Fabrication. *IEEE Transactions on Semiconductor Manufacturing, 22*, pp. 381-390.

Clark, A. (2005). Rolling horizon heuristics for production planning and set-up scheduling with backlogs and error-prone demand forecasts. *Production Planning & Control, 16*, pp. 81-97.

Clark, A., & Scarf, H. (1960). Optimal policies for the multi-echelon inventory problem. *Management Science, 6*, pp. 475–490.

Conway, R., Maxwell, W., & Miller, L. (1967). *Theory of Scheduling.* Reading, MA: Addison-Wesley.

Dobson, G., & Yano, C. A. (2002). Product offering, pricing, and make-to-stock/make-to-order decisions with shared capacity. *Production and Operations Management, 11*, pp. 293-312.

Domdouzis, K., Kumar, B., & Anumba, C. (2007). Radio-Frequency Identification (RFID) applications: A brief introduction. *Advanced Engineering Informatics, 21*, pp. 350-355.

Donk, D. P. (2001). Make to stock or make to order:The decoupling point in the food processing industries. *International Journal of Production Economics, 69*, pp. 297-306.

Duenyas, I., & Patananake, P. (1998). Base-stock control for single product tandem make-to-stock systems. *IIE Transactions, 30*, pp. 31–39.

Edwards, J. (1983). MRP and Kanban - American Style. *APICS 26th Conference*

*Proceedings*, pp. 586-603.

Eivazy, H., Rabbani, M., & Ebadian, M. (2009). A developed production control and scheduling model in the semiconductor manufacturing systems with hybrid make-to-stock/make-to-order products. *International Journal of Advanced Manufacturing Technology, 45*, pp. 968–986.

Elberse, A. (2008). Should you invest in the long tail? *Harvard Business Review, 86*, pp. 88-96.

Elias, S. (2002). *3 Day Car Programme Research Survey Report: New Car Buyer Behaviour.* Cardiff,UK:Cardiff University.

Elliott, G., & Timmermann, A. (2008). Economic Forecasting. *Journal of Economic Literature, 46*, pp. 3-56.

Ferrer, G., Heath, S. K., & Dew, N. (2011). An RFID application in large job shop remanufacturing operations. *International Journal Production Economics, 133*, pp. 612-621.

Fildes, R., & Kingsman, B. (2011). Incorporating demand uncertainty and forecast error in supply chain planning models. *Journal of the Operational Research Society, 62*, pp. 483-500.

Fildes, R., & Kumar, V. (2002). T elecommunications demand forecasting—a review. *International Journal of Forecasting, 18*, pp. 489–522.

Fildes, R., Nikolopoulos, K., Crone, S., & Syntetos, A. (2008). Forecasting and operational research: a review. *Journal of the Operational Research Society, 59*, pp. 1150-1172.

Fitzsimons, G. (2000). Consumer response to stockouts. *Journal of Consumer Research, 27*, pp. 249–266.

Fredriksson, P., & Gadde, L. (2005). Flexibility and rigidity in customization and build-to-order production. *Industrial Marketing Management, 34*, pp. 695-705.

Gail, T., & Scott, L. (1997). A field study investigating the effect of waiting time on customer satisfaction. *The Journal of Psychology, 131*, pp. 655-660.

Gershwin, S. (2000). Design and operating of manufacturing system-control point policy. *IIE Transactions, 32*, pp. 891–906.

Gershwin, S., Tan, B., & Veatch, M. (2009). Production control with backlog-dependent demand. *IIE Transactions, 41*, pp. 511–523.

Gfrerer, H., & Zapfel, G. (1995). Hierarchical model for production planning in the case of uncertain demand. *European Journal of Operational Research, 86*, pp. 142-161.

Goh, C., & Law, R. (2011). The Methodological Progress of Tourism Demand Forecasting: A Review of Related Literature. *Journal of Travel & Tourism Marketing, 28*, pp. 296-317.

Grabot, B., Geneste, L., Reynoso-Castillo, G., & Verot, S. (2005). Integration of uncertain and imprecise orders in the MRP method. *Journal of Intelligent Manufacturing, 16*, pp. 215–234.

Granger, C. W., & Hashem, P. M. (2000). Economic and Statistical Measures of Forecast Accuracy. *Journal of Forecasting, 19*, pp. 537-560.

Guillaume, R., Thierry, C., & Grabot, B. (2011). Modelling of ill-known requirements and integration in production planning. *Production Planning & Control, 22*, pp. 336-352.

Gunasekaran, A., & Ngai, E. W. (2009). Modeling and analysis of build-to-order supply chains. *European Journal of Operational Research*, 195, pp. 319–334.

Guo, R., Chiang, D., & Pai, F. (2007). A WIP-based exception-management model for integrated circuit back-end production processes. *International Journal of Advanced Manufacturing Technology, 33,* pp. 1263–1274.

Hallgren, M., & Olhager, J. (2006). Differentiating manufacturing focus. *International Journal of Production Research, 44(18),* pp. 3863-3878.

Haskose, A., Kingsman, B., & Worthington, D. (2004). Performance analysis of make-to-order manufacturing systems under different workload control regimes. *International Journal of Production Economics, 90(2),* pp. 169-186.

Hemmati, S., & Rabbani, M. (2010). Make-to-order/make-to-stock partitioning decision using the analytic network process. *International Journal of Advanced Manufacturing Technology, 48,* pp. 801–813.

Hino, S. (2006). *Inside the mind of Toyota : management principles for enduring growth.* New York, N.Y.: Productivity Press.

Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural Networks for Short-Term Load Forecasting:A Review and Evaluation. *IEEE Transactions on Power Systems, 16,* pp. 44-55.

Hoekstra, S., & Romme, J. (1992). *Integral Logistic Structures:Developing Customer-oriented Goods Flow.* London: McGraw-Hill.

Holweg, M., & Pil, F. (2001). Successful Build-to-Order Strategies start with the Customer. *MIT Sloan Management Review, 43,* pp. 74-83.

Holweg, M., & Pil, F. (2004). *The Second Century:Reconnecting Customer and Value Chain through Build-to-Order.* Cambridge,MA:MIT Press.

Homburg, C., & Fürst, A. (2005). How Organizational Complaint Handling Drives Customer Loyalty: An Analysis of the Mechanistic and the Organic Approach. *Journal of Marketing, 69,* pp. 95–114.

Homburg, C., & Giering, A. (2001). Personal characteristics as moderators of the relationship between customer satisfaction and loyalty - An empirical analysis. *Psychology & Marketing, 18(1),* pp. 43-66.

Hopp, W., & Roof, M. (1998). Setting WIP levels with statistical throughput control (STC) in CONWIP production lines. *International Journal of Production Research, 36(4),* pp. 867–882.

Hozak, K., & Hill, J. A. (2009). Issues and opportunities regarding replanning and rescheduling frequencies. *International Journal of Production Research, 47,* pp. 4955-4970.

Hu, H., Jiang, Z., & Zhang, H. (2010). A dynamic WIP control strategy for bottlenecks in a wafer fabrication system. *International Journal of Production Research, 48,* pp. 5221-5233.

Huang, G. Q., Zhang, Y., & Jiang, P. (2008). RFID-based wireless manufacturing for real-time management of job shop WIP inventories. *International Journal of Advanced Manufacturing Technology, 36,* pp. 752-764.

Hyer, N., & Wemmerlöv, U. (2002). *Reorganizing the factory : competing through cellular manufacturing.* Portland, Or.: Productivity Press.

Iravani, S. M., Liu, T., & Simchi-Levi, D. (2012). Optimal Production and Admission Policies in Make-to-Stock/Make-to-Order Manufacturing Systems. *Production and Operations Management, 21,* pp. 224–235.

Jamalnia, A., & Soukhakian, M. (2009). A hybrid fuzzy goal programming approach with different goal priorities. *Computers and Industrial Engineering, 56,* pp. 1474–1486.

Johnson, L., & Montgomery, D. (1974). *Operations Research in Production Planning, Scheduling and Inventory Control.* New York: Wiley.

Jonsson, P., & Mattsson, S. (2006). A longitudinal study of material planning applications in manufacturing companies. *International Journal of Operations and Production Management, 26*, pp. 971-995.

Kennedy, W., Patterson, J., & Fredendall, L. (2002). An overview of recent literature on spare parts inventories. *International Journal of Production Economics, 76(2)*, pp. 201-215.

Kim, J., & Leachman, R. (1994). Decomposition method application to a large scale linear programming WIP projection model. *European Journal of Operational Research, 74*, pp. 152–160.

Kimball, G. E. (1988). General principles of inventory control. *Journal of manufacturing and operations management, 1(1)*, pp. 119–130.

Koren, Y. (2010). *The global manufacturing revolution: product-process-business integration and reconfigurable systems.* Hoboken, N.J.: J. Wiley & Sons.

Kouvelis, P., Chambers, C., & Yu, D. (2005). Manufacturing operations manuscripts published in the first 52 issues of POM: Review, trends, and opportunities. *Production and Operations Management, 14*, pp. 450-467.

Kumar, P. (2005). The competitive impact of service process improvement: Examining customers' waiting experiences in retail markets. *Journal of Retailing, 81*, pp. 171–180.

Kumar, S., Nottestad, D. A., & Macklin, J. F. (2007). A profit and loss analysis for make-to-order versus make-to-stock policy:a supply chain case study. *Engineering Economist, 52(2)*, pp. 141-156.

Kuroda, M., & Kawada, A. (1995). Adaptive input control for job-shop type production systems with varying demands using inverse queuing network analysis. *International Journal of Production Economics, 41*, pp. 217–225.

Lan, Y., Liu, Y.-K., & Sun, G.-J. (2009). Modeling fuzzy multi-period production planning and sourcing problem with credibility service levels. *Journal of Computational and Applied Mathematics, 231*, pp. 208–221.

Lander, E., & Liker, J. K. (2007). The Toyota Production System and art: making highly customized and creative products the Toyota way. *International Journal of Production Research, 45(16)*, pp. 3681-3698.

Lee, H., Padmanabhan, V., & Whang, S. (1997). The bullwhip effect in supply chains. *Sloan Management Review, 38*, pp. 93–102.

Lee, J., Cho, Y., Lee, J.-D., & Lee, C.-Y. (2006). Forecasting future demand for large-screen television sets using conjoint analysis with diffusion model. *Technological Forecasting & Social Change, 73*, pp. 362–376.

Leitner, J., & Leopold-Wildburger, U. (2011). Experiments on forecasting behavior with several sources of information - A review of the literature. *European Journal of Operational Research, 213*, pp. 459-469.

Li, S., Tang, T., & Collins, D. (1996). Minimum inventory variability schedule with applications in semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing, 9*, pp. 145–149.

Liang, T. (2008). Fuzzy multi-objective production/distribution planning decisions with multi-product and multi-time period in a supply chain. *Computers and Industrial Engineering, 55*, pp. 676–694.

Lin, Y., & Lee, C. (2001). A total standard wip estimation method for wafer fabrication. *European Journal of Operational Research, 131*, pp. 78–94.

Luh, P., Zhou, X., & Tomastik, R. (2000). An effective method to reduce inventory in job shops. *IEEE Transactions on Robotics and Automation, 16*, pp. 420–424.

MacCarthy, B., & Fernandes, F. (2000). A multi-dimensional classification of production systems for the design and selection of production planning and control systems. *Production Planning & Control, 11(5)*, pp. 481-496.

Mallick, R., & Gaudreau, A. (1951). *Plant Layout.* New York: Wiley.

Mandelbaum, A., & Zeltyn, S. (2004). The impact of customers' patience on delay and abandonment:some empirically-driven experiments with the M/M/n+G queue. *OR Spectrum, 26*, pp. 377–411.

Meybodi, M., & Foote, B. (1995). Hierarchical production planning and scheduling with random demand and production failure. *Annals of Operations Research, 59*, pp. 259-280.

Miller, D. (1990). Simulation of a semiconductor manufacturing line. *Communications of the ACM, 33*, pp. 99–108.

Miltenburg, J. (2005). *Manufacturing strategy: how to formulate and implement a winning plan.* New York: Productivity Press.

Mula, J., Peidro, D., & Poler, R. (2010). The effectiveness of a fuzzy mathematical programming approach for supply chain production planning with fuzzy demand. *International Journal of Production Economics, 128*, pp. 136–143.

Mula, J., Poler, R., Garcia-Sabater, J., & Lario, F. (2006). Models for production planning under uncertainty: A review. *International Journal of Production Economics, 103*, pp. 271-285.

Mula, J., Polera, R., & Garcia-Sabater, J. (2007). Material requirement planning with fuzzy constraints and fuzzy coefficients. *Fuzzy Sets and Systems, 158*, pp. 783–793.

Nye, T., Jewkes, E., & Dilts, D. (2001). Optimal investment in setup reduction in manufacturing systems with WIP inventories. *Europen Journal of Operation Research, 135*, pp. 128–141.

Ohta, H., Hirota, T., & Rahim, A. (2007). Optimal production-inventory policy for make-to-order versus make-to-stock based on the M/Er/1 queuing model. *International Journal of Advanced Manufacturing Technology, 33*, pp. 36–41.

Olhager, J. (2003). Strategic positioning of order penetration point. *International Journal of Production Economics, 85*, pp. 319–329.

Papadopoulos, H., & Vidalis, M. (2001). Minimizing WIP inventory in reliable production lines. *International Journal of Production Economics, 70*, pp. 185-197.

Parry, G., & Graves, A. E. (2008). *Build to Order: The Road to the 5-Day Car.* London: Springer Verlag.

Peidro, D., Mula, J., Poler, R., & Verdegay, J.-L. (2009). Fuzzy optimization for supply chain planning under supply, demand and process uncertainties. *Fuzzy Sets and Systems, 160*, pp. 2640–2657.

Perona, M., Saccani, N., & Zanoni, S. (2009). Combining make-to-order and make-to-stock inventory policies: an empirical application to a manufacturing SME. *Production Planning & Control: The Management of Operations, 20(7)*, pp. 559-575.

Petrov, V. (1966). *Flowline Group Production Planning.* London: Business Publications.

Qi, C., Sivakumar, A. I., & Gershwin, S. B. (2008). Impact of production control and system factors in semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing, 21*, pp. 376-389.

Raa, B., & Aghezzaf, E. (2005). A robust dynamic planning strategy for lot-sizing problems with stochastic demands. *Journal of Intelligent Manufacturing, 16*, pp. 207–213.

Rabbani, S. H. (2010). Make-to-order/make-to-stock partitioning decision using the analytic network process. *International Journal of Advanced Manufacturing Technology, 48*, pp. 801–813.

Rafiei, H., & Rabbani, M. (2011). Order partitioning and Order Penetration Point location in hybrid Make-To-Stock/Make-To-Order production contexts. *Computers & Industrial Engineering, 61*, pp. 550–560.

Rafiei, H., & Rabbani, M. (2012). Capacity coordination in hybrid make-to-stock/make-to-order production environments. *International Journal of Production Research, 50(3)*, pp. 773-789.

Rajagopalan, S. (2002). Make to Order or Make to Stock: Model and Application. *Management Science, 48(2)*, pp. 241–256.

Robinson, L. W., & Chen, R. R. (2011). Estimating the Implied Value of the Customer's Waiting Time. *Manufacturing & Service Operations Management, 13*, pp. 53–57.

Ryan, S., & Choobineh, F. (2003). Total WIP and WIP mix for a CONWIP controlled job shop. *IIE Transactions, 35*, pp. 405–418.

Schultmann, F., Frohling, M., & Rentz, O. (2006). Fuzzy approach for production planning and detailed scheduling in paints manufacturing. *International Journal of Production Research, 44*, pp. 1589–1612.

Shao, X.-F., & Dong, M. (2012). Comparison of order-fulfilment performance in MTO and MTS systems with an inventory cost budget constraint. *International Journal of Production Research, 50(7)*, pp. 1917–1931.

Soman, C. A., Donk, D. P., & Gaalman, G. (2004). Combined make-to-order and make-to-stock in a food production system. *International Journal of Production Economics, 90*, pp. 223–235.

Soman, C. A., Donk, D. P., & Gaalman, G. (2006). Comparison of dynamic scheduling policies for hybrid make-to-order and make-to-stock production systems with stochastic demand. *International Journal of Production Economics, 104*, pp. 441–453.

Soman, C., Donk, D. v., & Gaalman, G. (2007). Capacitated planning and scheduling for combined make-to-order and make-to-stock production in the food industry: An illustrative case study. *International Journal of Production Economics, 108*, pp. 191–199.

Song, H., & Li, G. (2008). Tourism demand modelling and forecasting—A review of recent research. *Tourism Management, 29*, pp. 203–220.

Spearman, M., Woodruff, D., & Hopp, W. (1990). CONWIP: a pull alternative to kanban. *International Journal of Production Research, 28*, pp. 879–894.

Spitter, J., Kok, A. d., & Dellaert, N. (2005). Timing production in LP models in a rolling schedule. *Int. J. Production Economics, 93-94*, pp. 319-329.

Stevenson, M., Hendry, L. C., & Kingsman, B. G. (2005). A review of production planning and control: the applicability of key concepts to the make-to-order industry. *International Journal of Production Research, 43*, pp. 869–898.

Subramaniam, V., Yang, R., Chen, R., & Singhz, S. (2009). A WIP control policy for tandem lines. *International Journal of Production Research, 47(4)*, pp. 1127–1149.

Syntetos, A., Boylan, J., & Disney, S. (2009). Forecasting for inventory planning: a 50-year review. *Journal of the Operational Research Society, 60*, pp. S149-S160.

Taylor, S. (1994). Waiting for service: The relationship between delays and evaluations of service. *Journal of Marketing*, 58, pp.56-69

Teimoury, E., & Fathi, M. (2013). An integrated operations-marketing perspective for making decisions about order penetration point in multi-product supply chain: a queuing approach. *International Journal of Production Research, 51(18)*, pp. 5576-5596.

Thonemann, U. (2002). Improving supply-chain performance by sharing advance demand information. *European Journal of Operational Research, 142,* pp. 81-107.

Tolio, T., & Urgo, M. (2007). A Rolling Horizon Approach to Plan Outsourcing in Manufacturing-to-Order Environments Affected by Uncertainty. *Annals of the CIRP, 56*, pp. 487-490.

Tsai, K.-m., & Wang, S.-c. (2009). Multi-site available-to-promise modeling for assemble-to-order manufacturing: An illustration on TFT-LCD manufacturing. *International Journal of Production Economics, 117*, pp. 174–184.

Tsourveloudis, N., Doitsidis, L., & Ioannidis, S. (2007). Work-in-process scheduling by evolutionary tuned fuzzy controllers. *International Journal of Advanced Manufacturing Technology, 34*, pp. 748-761.

Tsubone, H., Ishikawa, Y., & Yamamoto, H. (2002). Production planning system for a combination of make-to-stock and make-to-order products. *International Journal of Production Research, 40*, pp. 4835-4851.

Uzsoy, R., Church, L. K., & Ovacik, I. M. (1992). Dispatching rules for semiconductor testing operations – a computational study. *Proceedings of IEEE/CHMT International Electronics Manufacturing Technology Symposium*.

Veatch, M. H. (2009). The impact of customer impatience on production control. *IIE Transactions, 41*, pp. 95–102.

Wargon, M., Guidet, B., Hoang, T. D., & Hejblum, G. (2009). A systematic review of models for forecasting the number of emergency department visits. *Emergency Medicine Journal, 26,* pp. 395-399.

Wee, B. v. (2007). Large infrastructure projects: a review of the quality of demand forecasts and cost estimations. *Environment and Planning B: Planning and Design, 34,* pp. 611-625.

Wein, L. (1988). Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing, 1*, pp. 115–130.

Wein, L. (1992). The relationship between yield and cycle time in semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing, 5*, pp. 156–158.

Wiendahl, H., ElMaraghy, H., & Nyhuis, P. (2007). Changeable manufacturing- Classification, design and operation. *CIRP Annals-Manufacturing Technology, 56*, pp. 783-809.

Wikipedia. (2013). *Build_to_stock.* Retrieved from wikipedia.org: http://en.wikipedia.org/wiki/Build_to_stock

Wikner, J., Naim, M. M., & Rudberg, M. (2007). Exploiting the order book for mass customized manufacturing control systems with capacity limitations. *IEEE Transactions on Engineering Management, 54(1),* pp. 145-155.

Wild, R. (1971). *The Techniques of Production Management.* London: Holt, Rinehart and Winston.

Wild, R. (1972). *Mass Production Management.* London: Wiley.

Womach, J., & Jones, D. (1996). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation.* New York: Simon and Schuster.

Womach, J., D.T.Jones, & D.Roos. (1990). *The Machine That Change the World: The Story of Lean Production.* New York: Scribner.

Woodward, J. (1980). *Industrial Organization- Theory and Practice* (2nd edition ed.). Oxford: Oxford University Press.

Wu, C.-H., Lin, J. T., & Wu, H.-H. (2010). Robust production and transportation planning in thin film transistor-liquid crystal display (TFT-LCD) industry under demand and price uncertainties. *International Journal of Production Research, 48*, pp. 6037-6060.

Wu, M.-C., Jiang, J.-H., & Chang, W.-J. (2008). Scheduling a hybrid MTO/MTS semiconductor fab with machine-dedication features. *International Journal of Production Economics, 112*, pp. 416–426.

Yingdong, M. (2001). *Design of hybrid make-to-stock (MTS)–make-to-order (MTO) manufacturing system.* MSc thesis,The University of Minnesota.

Zanjani, M., Nourelfath, M., & Ait-Kadi, D. (2010). A multi-stage stochastic programming approach for production planning with uncertainty in the quality of raw materials and demand. *International Journal of Production Research, 48*, pp. 4701-4723.

Zapfel, G. (1996). Production planning in the case of uncertain individual demand extension for an MRP II concept. *International Journal of Production Economics, 46-47*, pp. 153-164.

Zapfel, G. (1998). Customer-order-driven production: An economical concept for responding to demand uncertainty. *International Journal of Production Economics, 56-57*, pp. 699-709.

Zhang, X., Prajapati, M., & Peden, E. (2011). A stochastic production planning model under uncertain seasonal demand and market growth. *International Journal of Production Research, 49*, pp. 1957-1975.

Zhang, Z. G., Kim, I., Springer, M., Cai, G. G., & Yu, Y. (2013). Dynamic pooling of make-to-stock and make-to-order operations. *International Journal of Production Economics, http://dx.doi.org/10.1016/j.ijpe.2013.01.012i*.

## Appendix - A. MTB Program Source Code

This program was written by Microsoft Visual C# (2008 editing platform).

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace MTB_Identification
{
    public partial class Form1 : Form
    {

        int T, gama,Ta;
        double ifa, baita;
        double min, max;
        double lu, lo;
        double[] data;
        bool flag;
        Random ran;

        public Form1()
        {
            InitializeComponent();
            inital();
        }
         void inital()
        {
            data = new double[101];
        }

        void ValueAssign() //Assign the input value to parameters
        {
            Ta = 0;
            min = 0;

            T = int.Parse(Tt.Text);
            baita = double.Parse(m.Text);
            gama = int.Parse(c.Text);
            ifa = double.Parse(k.Text);

            lu= double.Parse(p1.Text);

            lo = double.Parse(c1.Text);

            ran = new Random(0);

            flag = false;
        }


        double ForecastError(int ta) //Function of forecast error
        {
            if (this.ifa < 0)
```

```
            return 0;
        else
            return 1 - Math.Exp(-this.ifa * ta);
    }


    double CustomerLoss(int tw) //Function of customer loss
    {
        if (this.baita * (tw - this.gama) > 1)
        {
            return 1;
        }
        else if (this.baita * (tw - this.gama) < 0)
        {
            return 0;
        }
        else
        {
            return this.baita * (tw - this.gama);
        }
    }

    double Loss(int ta) //Function to simulate the total loss per product
    {

        double z = 0,ra=ran.NextDouble();
        int tw = T - ta;
        double pf = ForecastError(ta);

        double pc = CustomerLoss(tw);

        if (pc >= pf*ra)
        {
            z = pc * lo + 0.5 *ra* pf * lu;
        }
        else
        {
            z = 0.5 *   (pc + ra*pf) * lo + 0.5 *ra* pf * lu;
        }

        return z;
    }

    double CalLoss(int ta) //Function to calculate the total loss per product
    {

        double z = 0, ra = 0.5;
        int tw = T - ta;
        double pf = ForecastError(ta);

        double pc = CustomerLoss(tw);

        if (pc >= pf*ra)
        {
            z = pc * lo + 0.5 * ra * pf * lu;
        }
        else
        {
            z = 0.5 * (pc + ra * pf) * lo + 0.5 * ra * pf * lu;
        }

        return z;
    }

    void DataToExcel() //export all data to the file "Interface.csv"
    {
        string path = @"MTB.csv";
```

```
        // Delete the file if it exists.
        if (File.Exists(path))
        {
            File.Delete(path);
        }

        try
        {
            FileStream fs = new FileStream(path, FileMode.OpenOrCreate);
            StreamWriter swWriter = new StreamWriter(fs);
            swWriter.WriteLine("Ta" + ',' + "Simlation" + ',' + "Calculation");
            for (int ta = 0; ta <= T; ta++)
            {

                double sum = 0;
                sum = 0;
                for (int i = 0; i < 1000; i++)
                {
                    sum += Loss(ta);
                }
                data[ta] = sum/1000;
                swWriter.WriteLine(ta.ToString() + ',' + data[ta].ToString() + ',' + CalLoss(ta).ToString());


            }

            swWriter.Close();
        }
        catch (Exception ee)
        {

            throw ee;
        }
    }

    void MarkUpandLowBound() //find the Max and Min value of loss
      {
           min = CalLoss(0);
           for (int ta = 1; ta <= T; ta++)
           {

                if (min > CalLoss(ta))
                {
                     min = CalLoss(ta);
                     Ta = ta;
                }
           }
           SimMinProfitLoss.Text = Decimal.Round((Decimal)min,2).ToString();
           SimOpTa.Text = Ta.ToString();

           max = CalLoss(0);
           for (int ta = 1; ta <= T; ta++)
           {

                if (max < CalLoss(ta))
                {
                     max = CalLoss(ta);
                     Ta = ta;
                }
           }
      }

      void CalculateOptimumTa()//Calculate the optimal Ta with given conditions
      {

           double ta = (this.ifa + 2 * this.baita - Math.Sqrt((this.ifa + 2 * this.baita) * (this.ifa + 2 * this.baita)
```

```
- 4 * this.ifa * this.ifa * this.baita * (this.T - this.gama))) / this.ifa / this.ifa;

            double lossZero = CalLoss(0);
            double lossTa = CalLoss((int)Round(ta, 0));
            double lossT = CalLoss((int)Round(this.T - this.gama, 0));

            if (lossTa < lossZero && lossTa < lossT)
            {
                CalMinProfitLoss.Text = Decimal.Round((Decimal)lossTa,2).ToString();
                CalOpTa.Text=Decimal.Round((Decimal)ta).ToString();
            }
            else if (lossZero < lossTa && lossZero < lossT)
            {
                CalOpTa.Text = "0";
                CalMinProfitLoss.Text= lossZero.ToString();
            }
            else if (lossT < lossTa && lossT < lossZero)
            {
                CalMinProfitLoss.Text = Decimal.Round((Decimal)lossT,2).ToString();
                CalOpTa.Text = (T - gama).ToString();
            }

        }

        private void Calculation_Click(object sender, EventArgs e)
        {
            ValueAssign();

            DataToExcel();
            DatatoMatix();
            CalculateOptimumTa();
            flag = true;

        }

        private void panel1_Paint(object sender, PaintEventArgs e) //Painting function
        {
            Graphics g = e.Graphics;
            Font vertFont = new Font("Verdana", 8, FontStyle.Bold);
            Font horzFont = new Font("Verdana", 8, FontStyle.Bold);
            SolidBrush vertBrush = new SolidBrush(Color.Blue);
            SolidBrush horzBrush = new SolidBrush(Color.Blue);
            Pen bluePen = new Pen(Color.Blue, 2);
            Pen blackPen = new Pen(Color.Black, 2);

            float fx = (float)(panel1.Size.Width);
            float fy = (float)(panel1.Size.Height);

            g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.95), (int)(fy * 0.9));
            g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.1), (int)(fy * 0.1));

            g.DrawString("0", horzFont, horzBrush, (int)(fx * 0.05), (int)(fy * 0.9));
            g.DrawString("10", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9));
            g.DrawString("20", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9));
            g.DrawString("30", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9));
            g.DrawString("40", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9));
            g.DrawString("50", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9));
            g.DrawString("60", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9));
            g.DrawString("70", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9));
            g.DrawString("80", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9));
            g.DrawString("90", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9));
            g.DrawString("100", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9));

            StringFormat vertStrFormat = new StringFormat();
            vertStrFormat.FormatFlags = StringFormatFlags.DirectionVertical;

            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9) - 8, vertStrFormat);
```

```
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9) - 8, vertStrFormat);
        g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9) - 8, vertStrFormat);

        g.DrawString("1 -", vertFont, vertBrush, (int)(fx * 0.04), (int)(fy * 0.15));


        vertFont.Dispose();
        vertBrush.Dispose();
        horzFont.Dispose();
        horzBrush.Dispose();
        blackPen.Dispose();
        bluePen.Dispose();
        g.Dispose();
    }


    private void Panel1_MouseClick(object sender, MouseEventArgs e) //Painting function
    {
        Graphics g = panel1.CreateGraphics();
        g.Clear(Color.WhiteSmoke);
        if (flag)
        {

            Font vertFont = new Font("Verdana", 8, FontStyle.Bold);
            Font horzFont = new Font("Verdana", 8, FontStyle.Bold);
            SolidBrush vertBrush = new SolidBrush(Color.Blue);
            SolidBrush horzBrush = new SolidBrush(Color.Blue);
            Pen bluePen = new Pen(Color.Blue, 2);
            Pen blackPen = new Pen(Color.Black, 2);

            float fx = (float)(panel1.Size.Width);
            float fy = (float)(panel1.Size.Height);

            g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.95), (int)(fy * 0.9));
            g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.1), (int)(fy * 0.1));

            g.DrawString("0", horzFont, horzBrush, (int)(fx * 0.05), (int)(fy * 0.9));
            g.DrawString("10", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9));
            g.DrawString("20", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9));
            g.DrawString("30", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9));
            g.DrawString("40", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9));
            g.DrawString("50", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9));
            g.DrawString("60", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9));
            g.DrawString("70", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9));
            g.DrawString("80", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9));
            g.DrawString("90", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9));
            g.DrawString("100", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9));

            StringFormat vertStrFormat = new StringFormat();
            vertStrFormat.FormatFlags = StringFormatFlags.DirectionVertical;

            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9) - 8, vertStrFormat);
```

```
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9) - 8, vertStrFormat);

                g.DrawString("1 -", vertFont, vertBrush, (int)(fx * 0.04), (int)(fy * 0.15));

                Pen linePen = new Pen(Color.Black, 2);

                Point startPoint, endPoint;
                startPoint = new Point((int)(fx * 0.1), (int)(fy * (0.9 - ForecastError(0) * 0.75)));
                for (int ta = 1; ta <= T; ta++)
                {
                    endPoint = new Point((int)(fx * (0.1 + ta / 10.0 * 0.085)), (int)(fy * (0.9 -
ForecastError(ta) * 0.75)));
                    g.DrawLine(linePen, startPoint, endPoint);
                    startPoint = endPoint;
                }
                linePen.Dispose();
                vertFont.Dispose();
                vertBrush.Dispose();
                horzFont.Dispose();
                horzBrush.Dispose();
                blackPen.Dispose();
                bluePen.Dispose();
            }
            g.Dispose();
        }



        private void panel2_Paint(object sender, PaintEventArgs e) //Painting function
        {
            Graphics g = e.Graphics;
            Font vertFont = new Font("Verdana", 8, FontStyle.Bold);
            Font horzFont = new Font("Verdana", 8, FontStyle.Bold);
            SolidBrush vertBrush = new SolidBrush(Color.Blue);
            SolidBrush horzBrush = new SolidBrush(Color.Blue);
            Pen bluePen = new Pen(Color.Blue, 2);
            Pen blackPen = new Pen(Color.Black, 2);

            float fx = (float)(panel2.Size.Width);
            float fy = (float)(panel2.Size.Height);

            g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.95), (int)(fy * 0.9));
            g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.1), (int)(fy * 0.1));

            g.DrawString("0", horzFont, horzBrush, (int)(fx * 0.05), (int)(fy * 0.9));
            g.DrawString("10", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9));
            g.DrawString("20", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9));
            g.DrawString("30", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9));
            g.DrawString("40", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9));
            g.DrawString("50", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9));
            g.DrawString("60", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9));
            g.DrawString("70", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9));
            g.DrawString("80", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9));
            g.DrawString("90", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9));
            g.DrawString("100", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9));

            StringFormat vertStrFormat = new StringFormat();
            vertStrFormat.FormatFlags = StringFormatFlags.DirectionVertical;

            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9) - 8, vertStrFormat);
```

```
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9) - 8, vertStrFormat);

            g.DrawString("1 -", vertFont, vertBrush, (int)(fx * 0.04), (int)(fy * 0.15));


            vertFont.Dispose();
            vertBrush.Dispose();
            horzFont.Dispose();
            horzBrush.Dispose();
            blackPen.Dispose();
            bluePen.Dispose();
            g.Dispose();
        }

        private void panel2_MouseClick(object sender, MouseEventArgs e) //Painting function
        {
            Graphics g = panel2.CreateGraphics();
            g.Clear(Color.WhiteSmoke);
            if (flag)
            {
                Font vertFont = new Font("Verdana", 8, FontStyle.Bold);
                Font horzFont = new Font("Verdana", 8, FontStyle.Bold);
                SolidBrush vertBrush = new SolidBrush(Color.Blue);
                SolidBrush horzBrush = new SolidBrush(Color.Blue);
                Pen bluePen = new Pen(Color.Blue, 2);
                Pen blackPen = new Pen(Color.Black, 2);

                float fx = (float)(panel2.Size.Width);
                float fy = (float)(panel2.Size.Height);

                g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.95), (int)(fy * 0.9));
                g.DrawLine(bluePen, (int)(fx * 0.1), (int)(fy * 0.9), (int)(fx * 0.1), (int)(fy * 0.1));

                g.DrawString("0", horzFont, horzBrush, (int)(fx * 0.05), (int)(fy * 0.9));
                g.DrawString("10", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9));
                g.DrawString("20", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9));
                g.DrawString("30", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9));
                g.DrawString("40", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9));
                g.DrawString("50", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9));
                g.DrawString("60", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9));
                g.DrawString("70", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9));
                g.DrawString("80", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9));
                g.DrawString("90", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9));
                g.DrawString("100", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9));

                StringFormat vertStrFormat = new StringFormat();
                vertStrFormat.FormatFlags = StringFormatFlags.DirectionVertical;

                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.135), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.22), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.305), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.385), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.47), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.555), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.64), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.725), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.81), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.895), (int)(fy * 0.9) - 8, vertStrFormat);

                g.DrawString("1 -", vertFont, vertBrush, (int)(fx * 0.04), (int)(fy * 0.15));


                vertFont.Dispose();
                vertBrush.Dispose();
                horzFont.Dispose();
                horzBrush.Dispose();
```

```
                blackPen.Dispose();
                bluePen.Dispose();

                Pen linePen = new Pen(Color.Black, 2);

                Point startPoint, endPoint;
                startPoint = new Point((int)(fx * 0.08), (int)(fy * (0.9 - CustomerLoss(0) * 0.75)));
                for (int ta = 1; ta <= T; ta++)
                {
                    endPoint = new Point((int)(fx * (0.08 + ta / 10.0 * 0.085)), (int)(fy * (0.9 -
CustomerLoss(ta) * 0.75)));
                    g.DrawLine(linePen, startPoint, endPoint);
                    startPoint = endPoint;
                }
                linePen.Dispose();
                g.Dispose();
            }
        }


        private void Panel3_Paint(object sender, PaintEventArgs e) //Painting function
        {
            Graphics g = e.Graphics;
            Font vertFont = new Font("Verdana", 10, FontStyle.Bold);
            Font horzFont = new Font("Verdana", 10, FontStyle.Bold);
            SolidBrush vertBrush = new SolidBrush(Color.Blue);
            SolidBrush horzBrush = new SolidBrush(Color.Blue);
            Pen bluePen = new Pen(Color.Blue, 2);
            Pen blackPen = new Pen(Color.Black, 2);

            float fx = (float)(panel3.Size.Width);
            float fy = (float)(panel3.Size.Height);

            g.DrawLine(bluePen, (int)(fx * 0.065), (int)(fy * 0.9), (int)(fx * 0.95), (int)(fy * 0.9));
            g.DrawLine(bluePen, (int)(fx * 0.065), (int)(fy * 0.9), (int)(fx * 0.065), (int)(fy * 0.1));


            g.DrawString("0", horzFont, horzBrush, (int)(fx * 0.04), (int)(fy * 0.9));
            g.DrawString("10", horzFont, horzBrush, (int)(fx * 0.1285), (int)(fy * 0.9));
            g.DrawString("20", horzFont, horzBrush, (int)(fx * 0.217), (int)(fy * 0.9));
            g.DrawString("30", horzFont, horzBrush, (int)(fx * 0.3055), (int)(fy * 0.9));
            g.DrawString("40", horzFont, horzBrush, (int)(fx * 0.394), (int)(fy * 0.9));
            g.DrawString("50", horzFont, horzBrush, (int)(fx * 0.4825), (int)(fy * 0.9));
            g.DrawString("60", horzFont, horzBrush, (int)(fx * 0.571), (int)(fy * 0.9));
            g.DrawString("70", horzFont, horzBrush, (int)(fx * 0.6595), (int)(fy * 0.9));
            g.DrawString("80", horzFont, horzBrush, (int)(fx * 0.748), (int)(fy * 0.9));
            g.DrawString("90", horzFont, horzBrush, (int)(fx * 0.8365), (int)(fy * 0.9));
            g.DrawString("100", horzFont, horzBrush, (int)(fx * 0.925), (int)(fy * 0.9));

            StringFormat vertStrFormat = new StringFormat();
            vertStrFormat.FormatFlags = StringFormatFlags.DirectionVertical;

            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.1285), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.217), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.3055), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.394), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.4825), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.571), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.6595), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.748), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.8365), (int)(fy * 0.9) - 8, vertStrFormat);
            g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.925), (int)(fy * 0.9) - 8, vertStrFormat);


            g.DrawString("30 -", vertFont, vertBrush, 0, (int)(fy * 0.15));
            g.DrawString("25 -", vertFont, vertBrush, 0, (int)(fy * 0.27));
            g.DrawString("20 -", vertFont, vertBrush, 0, (int)(fy * 0.39));
```

```
        g.DrawString("15 -", vertFont, vertBrush, 0, (int)(fy * 0.51));
        g.DrawString("10 -", vertFont, vertBrush, 0, (int)(fy * 0.63));
        g.DrawString(" 5 -", vertFont, vertBrush, 0, (int)(fy * 0.75));


        vertFont.Dispose();
        vertBrush.Dispose();
        horzFont.Dispose();
        horzBrush.Dispose();
        blackPen.Dispose();
        bluePen.Dispose();
        g.Dispose();
}

private void Panel3_MouseClick(object sender, MouseEventArgs e) //Painting function
{
         Graphics g = panel3.CreateGraphics();
         g.Clear(Color.WhiteSmoke);
        if (flag)
        {

                Font vertFont = new Font("Verdana", 10, FontStyle.Bold);
                Font horzFont = new Font("Verdana", 10, FontStyle.Bold);
                SolidBrush vertBrush = new SolidBrush(Color.Blue);
                SolidBrush horzBrush = new SolidBrush(Color.Blue);
                Pen bluePen = new Pen(Color.Blue, 2);
                Pen blackPen = new Pen(Color.Black, 2);

                float fx = (float)(panel3.Size.Width);
                float fy = (float)(panel3.Size.Height);

                g.DrawLine(bluePen, (int)(fx * 0.065), (int)(fy * 0.9), (int)(fx * 0.95), (int)(fy * 0.9));
                g.DrawLine(bluePen, (int)(fx * 0.065), (int)(fy * 0.9), (int)(fx * 0.065), (int)(fy * 0.1));

                g.DrawString("0", horzFont, horzBrush, (int)(fx * 0.04), (int)(fy * 0.9));
                g.DrawString("10", horzFont, horzBrush, (int)(fx * 0.1285), (int)(fy * 0.9));
                g.DrawString("20", horzFont, horzBrush, (int)(fx * 0.217), (int)(fy * 0.9));
                g.DrawString("30", horzFont, horzBrush, (int)(fx * 0.3055), (int)(fy * 0.9));
                g.DrawString("40", horzFont, horzBrush, (int)(fx * 0.394), (int)(fy * 0.9));
                g.DrawString("50", horzFont, horzBrush, (int)(fx * 0.4825), (int)(fy * 0.9));
                g.DrawString("60", horzFont, horzBrush, (int)(fx * 0.571), (int)(fy * 0.9));
                g.DrawString("70", horzFont, horzBrush, (int)(fx * 0.6595), (int)(fy * 0.9));
                g.DrawString("80", horzFont, horzBrush, (int)(fx * 0.748), (int)(fy * 0.9));
                g.DrawString("90", horzFont, horzBrush, (int)(fx * 0.8365), (int)(fy * 0.9));
                g.DrawString("100", horzFont, horzBrush, (int)(fx * 0.925), (int)(fy * 0.9));

                StringFormat vertStrFormat = new StringFormat();
                vertStrFormat.FormatFlags = StringFormatFlags.DirectionVertical;

                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.1285), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.217), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.3055), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.394), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.4825), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.571), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.6595), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.748), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.8365), (int)(fy * 0.9) - 8, vertStrFormat);
                g.DrawString("-", horzFont, horzBrush, (int)(fx * 0.925), (int)(fy * 0.9) - 8, vertStrFormat);


                double h = (Math.Floor(max / 3) + 1) * 3;
                double tag = h / 6;

                g.DrawString((h).ToString()+ " -", vertFont, vertBrush, 0, (int)(fy * 0.15));
                g.DrawString((h-tag ).ToString() + " -", vertFont, vertBrush, 0, (int)(fy * 0.27));
                g.DrawString((h - 2 * tag).ToString() + " -", vertFont, vertBrush, 0, (int)(fy * 0.39));
```

```
                                        g.DrawString((h - 3 * tag).ToString() + " -", vertFont, vertBrush, 0, (int)(fy * 0.51));
                                        g.DrawString((h - 4 * tag).ToString() + " -", vertFont, vertBrush, 0, (int)(fy * 0.63));
                                        g.DrawString((h - 5 * tag).ToString() + " -", vertFont, vertBrush, 0, (int)(fy * 0.75));



                                        vertFont.Dispose();
                                        vertBrush.Dispose();
                                        horzFont.Dispose();
                                        horzBrush.Dispose();
                                        blackPen.Dispose();
                                        bluePen.Dispose();

                                        Pen linePen = new Pen(Color.Black, 2);

                                        Point startPoint, endPoint;
                                        startPoint = new Point((int)(fx * 0.065), (int)(fy * (0.9-(data[0])/tag*0.12)));
                                        for (int ta = 1; ta <= T; ta++)
                                        {
                                                endPoint = new Point((int)(fx * (0.065+ta/10.0*0.0885)), (int)(fy *
(0.9-(data[ta])/tag*0.12)));
                                                g.DrawLine(linePen, startPoint, endPoint);
                                                startPoint = endPoint;
                                        }
                                        linePen.Dispose();

                                        Pen CalPen = new Pen(Color.Red, 2);


                                        startPoint = new Point((int)(fx * 0.065), (int)(fy * (0.9 - (data[0]) / tag * 0.12)));
                                        for (int ta = 1; ta <= T; ta++)
                                        {
                                                endPoint = new Point((int)(fx * (0.065 + ta / 10.0 * 0.0885)), (int)(fy * (0.9 -
(CalLoss(ta)) / tag * 0.12)));
                                                g.DrawLine(CalPen, startPoint, endPoint);
                                                startPoint = endPoint;
                                        }
                                        CalPen.Dispose();
                                        g.Dispose();
                                }
                        }


                        double Round(double value, int decimals)//Rounds a number to a specified number of decimal places
or to a whole number if no decimal places are specified
                        {
                                if (value < 0)
                                {
                                        return Math.Round(value + 5 / Math.Pow(10, decimals + 1), decimals,
MidpointRounding.AwayFromZero);
                                }
                                else
                                {
                                        return Math.Round(value, decimals, MidpointRounding.AwayFromZero);
                                }
                        }

                }
}
```

## Appendix - B. Smart MTB Program Source Code

The following program code was written by Microsoft Visual C# (2008 editing platform).

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace OptimalTaCalculation
{
    public partial class Form1 : Form
    {
        public int T;
        public double ifa;
        public double baita;
        public double gama;
        public double lo;
        public double lu;
        public int Tunit;
        public int ifaunit;
        public int baitaunit;
        public int gamaunit;
        public int profitunit;
        public int lossunit;

        public void init()// Initialize the parameters with input values;
        {
            Tunit = Timeunit.SelectedIndex;
            baitaunit = CLunit.SelectedIndex;
            gamaunit = WTunit.SelectedIndex;
            profitunit = Punit.SelectedIndex;
            lossunit = Lunit.SelectedIndex;

            T = int.Parse(Time.Text);
            ifa = - Math.Log(1- double.Parse(FE.Text)/100)/T;
            baita = double.Parse(CL.Text)/100;
            gama = double.Parse(WT.Text);
            lu = double.Parse(profit.Text);
            lo = double.Parse(loss.Text);

        }

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
```

```
        {
            init();
            if (Tunit == baitaunit && baitaunit == gamaunit && gamaunit != -1 && profitunit == lossunit
&& lossunit != -1)
            {
                CalculateOptimumTa();
            }
            else
            {
                System.Windows.Forms.MessageBox.Show("Please select correct parameter unit",
"Error",MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

            }

        }


        void CalculateOptimumTa() //Calculate the optimal Ta with given conditions
        {

            double s3 = (this.ifa + this.baita - Math.Sqrt((this.ifa + this.baita) * (this.ifa + this.baita) - 2 *
this.ifa * this.ifa * this.baita * (this.T - this.gama))) / this.ifa / this.ifa;

            string Taunit;
            if (Tunit == 0)
            {
                Taunit = " week";
            }
            else if (Tunit == 1)
            {
                Taunit = " day";
            }
            else
            {
                Taunit = " hour";
            }



            if (Loss((int)Round(s3, 0)) < Loss(0) && Loss((int)Round(s3, 0)) < Loss((int)Round(this.T -
this.gama, 0)))
            {
                result.Text = (Round(s3,0)).ToString() + Taunit + "; Loss:" +
Decimal.Round((Decimal)Loss((int)Round(s3, 0)), 2).ToString() + Lunit.Text+" per product";

            }
            else if (Loss(0) < Loss((int)Round(s3, 0)) && Loss(0) < Loss((int)Round(this.T - this.gama, 0)))
            {
                result.Text = "0"+ "; Loss:" +Decimal.Round((Decimal)Loss(0),2).ToString() + Lunit.Text+"
per product";

            }
            else if (Loss((int)Round(this.T - this.gama, 0)) < Loss((int)Round(s3, 0)) &&
Loss((int)Round(this.T - this.gama, 0)) < Loss(0))
            {
                result.Text = (T - gama).ToString() + Taunit + "; Loss:" +
Decimal.Round((Decimal)Loss((int)Round(this.T - this.gama, 0)),2).ToString() + Lunit.Text+" per product";

            }

         }



        double Loss(int ta)//calculate the total loss per product
        {
             double z = 0, ra = 0.5;
```

```
            int tw = T - ta;
            double pf = ForecastError(ta);

            double pc = CustomerLoss(tw);

            if (pc >= pf * ra)
            {
                z = pc * lo + 0.5 * ra * pf * lu;
            }
            else
            {
                z = 0.5 * (pc + ra * pf) * lo + 0.5 * ra * pf * lu;
            }
            return z;
        }

        double ForecastError(int ta)//forecast error
        {
            if (this.ifa == 0)
                return 0;
            else
                return 1 - Math.Exp(-this.ifa * ta);
        }

        double CustomerLoss(double tw)//customer loss percentage
       {
            if (this.baita * (tw - this.gama) > 1)
            {
                return 1;
            }
            else if ((tw - this.gama) < 0)
            {
                return 0;
            }
            else
            {
                return this.baita * (tw - this.gama);
            }
        }


        double Round(double value, int decimals)
//Rounds a number to a specified number of decimal places or to a whole number if no decimal places are
specified
        {
            if (value < 0)
            {
                return Math.Round(value + 5 / Math.Pow(10, decimals + 1), decimals,
MidpointRounding.AwayFromZero);
            }
            else
            {
                return Math.Round(value, decimals, MidpointRounding.AwayFromZero);
            }
        }

        private void TimeChanged(object sender, EventArgs e)
        {
            textBox1.Text = Time.Text + " " + Timeunit.Text;
        }

        private void UnitChanged(object sender, EventArgs e)
        {
            CLunit.Text = Timeunit.Text;
            WTunit.Text=Timeunit.Text;
        }
```

```
        private void CLunitChanged(object sender, EventArgs e)
        {
            Timeunit.Text = CLunit.Text;
            WTunit.Text = CLunit.Text;

        }

        private void WTunitChanged(object sender, EventArgs e)
        {
            CLunit.Text=WTunit.Text;
            Timeunit.Text=WTunit.Text;

        }

        private void PunitChanged(object sender, EventArgs e)
        {
            Lunit.Text = Punit.Text;
        }

        private void LunitChanged(object sender, EventArgs e)
        {
            Punit.Text = Lunit.Text;
        }
    }
}
```

## Appendix - C. Simulation and Calculation Results in MTB Testing

With the parameters given in Table 5-1, the losses of simulation and calculation at different $T_a$ are listed as follows (simulation adopts random function to represent the forecast error, and the simulation result is the average value of 1000 time random calculations):

| $T_a$ | Program Simulation | Calculation | STELLA |
|---|---|---|---|
| 0 | 5.225 | 5.225 | 5.22 |
| 1 | 5.207638816 | 5.208424641 | 5.207533 |
| 2 | 5.191680322 | 5.19176369 | 5.1902 |
| 3 | 5.175955575 | 5.175017338 | 5.172467 |
| 4 | 5.156922604 | 5.158185776 | 5.155 |
| 5 | 5.139442211 | 5.141269193 | 5.137733 |
| 6 | 5.127316533 | 5.124267778 | 5.119667 |
| 7 | 5.105802162 | 5.107181721 | 5.1024 |
| 8 | 5.089471452 | 5.09001121 | 5.085 |
| 9 | 5.074604623 | 5.072756434 | 5.067467 |
| 10 | 5.044668736 | 5.055417579 | 5.0498 |
| 11 | 5.033416816 | 5.037994834 | 5.032733 |
| 12 | 5.020225555 | 5.020488385 | 5.0154 |
| 13 | 5.00416952 | 5.002898418 | 4.997933 |
| 14 | 4.978679514 | 4.98522512 | 4.9808 |
| 15 | 4.987905551 | 4.967468676 | 4.963333 |
| 16 | 4.944103281 | 4.949629271 | 4.942933 |
| 17 | 4.922346441 | 4.931707091 | 4.925133 |
| 18 | 4.903298234 | 4.913702319 | 4.9076 |
| 19 | 4.883553625 | 4.895615139 | 4.890067 |
| 20 | 4.885879097 | 4.877445736 | 4.8698 |
| 21 | 4.87565686 | 4.859194292 | 4.852 |
| 22 | 4.839249025 | 4.840860991 | 4.835067 |
| 23 | 4.808564127 | 4.822446013 | 4.8144 |
| 24 | 4.810572882 | 4.803949542 | 4.797 |
| 25 | 4.782056438 | 4.785371759 | 4.776267 |
| 26 | 4.759628018 | 4.766712844 | 4.759133 |
| 27 | 4.775864476 | 4.74797298 | 4.7382 |
| 28 | 4.726394161 | 4.729152345 | 4.7212 |
| 29 | 4.718818436 | 4.71025112 | 4.700533 |
| 30 | 4.706552859 | 4.691269485 | 4.683067 |
| 31 | 4.709094411 | 4.672207618 | 4.662467 |
| 32 | 4.657199808 | 4.653065698 | 4.645733 |
| 33 | 4.612768772 | 4.633843904 | 4.624533 |
| 34 | 4.642759948 | 4.614542414 | 4.604067 |

| | | | |
|---|---|---|---|
| 35 | 4.626017668 | 4.595161404 | 4.587 |
| 36 | 4.55773029 | 4.575701053 | 4.5662 |
| 37 | 4.590603329 | 4.556161536 | 4.546067 |
| 38 | 4.551558674 | 4.536543031 | 4.525467 |
| 39 | 4.528021561 | 4.516845713 | 4.508 |
| 40 | 4.522887293 | 4.497069757 | 4.487467 |
| 41 | 4.494610078 | 4.47721534 | 4.4668 |
| 42 | 4.412308632 | 4.457282635 | 4.446 |
| 43 | 4.404301304 | 4.437271818 | 4.4258 |
| 44 | 4.44254846 | 4.417183061 | 4.408467 |
| 45 | 4.39337987 | 4.397016539 | 4.388067 |
| 46 | 4.397664735 | 4.376772425 | 4.3674 |
| 47 | 4.394266482 | 4.356450891 | 4.346933 |
| 48 | 4.359331986 | 4.336052111 | 4.327067 |
| 49 | 4.335673933 | 4.315576256 | 4.3062 |
| 50 | 4.264197708 | 4.295023498 | 4.285733 |
| 51 | 4.301510859 | 4.274394009 | 4.2646 |
| 52 | 4.283687531 | 4.253687958 | 4.2444 |
| 53 | 4.262613895 | 4.232905517 | 4.223933 |
| 54 | 4.203751933 | 4.212046856 | 4.203533 |
| 55 | 4.195082393 | 4.191112145 | 4.1802 |
| 56 | 4.164039094 | 4.170101552 | 4.1618 |
| 57 | 4.188047557 | 4.149015248 | 4.145867 |
| 58 | 4.186372855 | 4.1278534 | 4.131667 |
| 59 | 4.115262952 | 4.106616178 | 4.1204 |
| 60 | 4.131991122 | 4.085303748 | 4.111933 |
| 61 | 4.195693304 | 4.063916279 | 4.0998 |
| 62 | 4.069487418 | 4.042453937 | 4.093133 |
| 63 | 4.120200184 | 4.02091689 | 4.086933 |
| 64 | 4.17103012 | 3.999305303 | 4.0788 |
| 65 | 4.149533007 | 3.977619343 | 4.0748 |
| 66 | 4.009013501 | 3.955859176 | 4.071067 |
| 67 | 4.066513993 | 3.934024967 | 4.066133 |
| 68 | 4.091755812 | 3.943480233 | 4.065133 |
| 69 | 4.045206351 | 3.959525975 | 4.063733 |
| 70 | 4.195115781 | 3.975474719 | 4.060667 |
| 71 | 4.03015057 | 3.99132668 | 4.061867 |
| 72 | 4.098140496 | 4.007082074 | 4.062333 |
| 73 | 4.143400049 | 4.022741115 | 4.0676 |
| 74 | 4.18689208 | 4.03830402 | 4.069867 |
| 75 | 4.174097851 | 4.053771001 | 4.077267 |
| 76 | 4.114569192 | 4.069142272 | 4.079933 |
| 77 | 4.135653013 | 4.084418047 | 4.085067 |
| 78 | 4.144197766 | 4.099598539 | 4.095133 |
| 79 | 4.17198983 | 4.114683958 | 4.100867 |
| 80 | 4.176861033 | 4.129674519 | 4.107467 |
| 81 | 4.219223562 | 4.14457043 | 4.12 |
| 82 | 4.145612965 | 4.159371905 | 4.128133 |
| 83 | 4.217933253 | 4.174079152 | 4.1376 |

| | | | |
|---|---|---|---|
| 84 | 4.243515299 | 4.188692382 | 4.1512 |
| 85 | 4.26654244 | 4.203211804 | 4.1608 |
| 86 | 4.166128382 | 4.217637627 | 4.1706 |
| 87 | 4.140980684 | 4.231970059 | 4.181467 |
| 88 | 4.259276825 | 4.246209309 | 4.193133 |
| 89 | 4.202724721 | 4.260355585 | 4.21 |
| 90 | 4.106034127 | 4.274409092 | 4.222267 |
| 91 | 4.364511898 | 4.288370039 | 4.234933 |
| 92 | 4.232627537 | 4.302238631 | 4.2484 |
| 93 | 4.36027657 | 4.316015073 | 4.2618 |
| 94 | 4.367972849 | 4.329699572 | 4.275667 |
| 95 | 4.35209645 | 4.343292331 | 4.290667 |
| 96 | 4.485468672 | 4.384293556 | 4.331 |
| 97 | 4.360095823 | 4.42520345 | 4.371733 |
| 98 | 4.422545257 | 4.466022216 | 4.411933 |
| 99 | 4.412789178 | 4.506750058 | 4.452667 |
| 100 | 4.468658692 | 4.547387179 | 4.493133 |

# Appendix - D. Simulation and Calculation Results in MTB Handling Extreme Situations

With the given parameters in **Table 5-1** and the four extreme situations in **Table 5-2** , the losses at different $T_a$ are listed as follows (simulation adopts random function to represent the forecast error, and the simulation result is the average value of 1000 time random calculations):

1. Totally accurate ($\alpha$ =0)

| $T_a$ | Program Simulation | Calculation | STELLA |
|---|---|---|---|
| 0 | 5.225 | 5.225 | 5. 22 |
| 1 | 5.17 | 5.17 | |
| 2 | 5.115 | 5.115 | |
| 3 | 5.06 | 5.06 | |
| 4 | 5.005 | 5.005 | |
| 5 | 4.95 | 4.95 | |
| 6 | 4.895 | 4.895 | |
| 7 | 4.84 | 4.84 | |
| 8 | 4.785 | 4.785 | |
| 9 | 4.73 | 4.73 | |
| 10 | 4.675 | 4.675 | |
| 11 | 4.62 | 4.62 | |
| 12 | 4.565 | 4.565 | |
| 13 | 4.51 | 4.51 | |
| 14 | 4.455 | 4.455 | |
| 15 | 4.4 | 4.4 | |
| 16 | 4.345 | 4.345 | |
| 17 | 4.29 | 4.29 | |
| 18 | 4.235 | 4.235 | |
| 19 | 4.18 | 4.18 | |
| 20 | 4.125 | 4.125 | 4. 13 |
| 21 | 4.07 | 4.07 | |
| 22 | 4.015 | 4.015 | |
| 23 | 3.96 | 3.96 | |
| 24 | 3.905 | 3.905 | |
| 25 | 3.85 | 3.85 | |
| 26 | 3.795 | 3.795 | |
| 27 | 3.74 | 3.74 | |

| | | | |
|---|---|---|---|
| 28 | 3.685 | 3.685 | |
| 29 | 3.63 | 3.63 | |
| 30 | 3.575 | 3.575 | |
| 31 | 3.52 | 3.52 | |
| 32 | 3.465 | 3.465 | |
| 33 | 3.41 | 3.41 | |
| 34 | 3.355 | 3.355 | |
| 35 | 3.3 | 3.3 | |
| 36 | 3.245 | 3.245 | |
| 37 | 3.19 | 3.19 | |
| 38 | 3.135 | 3.135 | |
| 39 | 3.08 | 3.08 | |
| 40 | 3.025 | 3.025 | 3.03 |
| 41 | 2.97 | 2.97 | |
| 42 | 2.915 | 2.915 | |
| 43 | 2.86 | 2.86 | |
| 44 | 2.805 | 2.805 | |
| 45 | 2.75 | 2.75 | |
| 46 | 2.695 | 2.695 | |
| 47 | 2.64 | 2.64 | |
| 48 | 2.585 | 2.585 | |
| 49 | 2.53 | 2.53 | |
| 50 | 2.475 | 2.475 | |
| 51 | 2.42 | 2.42 | |
| 52 | 2.365 | 2.365 | |
| 53 | 2.31 | 2.31 | |
| 54 | 2.255 | 2.255 | |
| 55 | 2.2 | 2.2 | |
| 56 | 2.145 | 2.145 | |
| 57 | 2.09 | 2.09 | |
| 58 | 2.035 | 2.035 | |
| 59 | 1.98 | 1.98 | |
| 60 | 1.925 | 1.925 | 1.93 |
| 61 | 1.87 | 1.87 | |
| 62 | 1.815 | 1.815 | |
| 63 | 1.76 | 1.76 | |
| 64 | 1.705 | 1.705 | |
| 65 | 1.65 | 1.65 | |
| 66 | 1.595 | 1.595 | |
| 67 | 1.54 | 1.54 | |
| 68 | 1.485 | 1.485 | |
| 69 | 1.43 | 1.43 | |
| 70 | 1.375 | 1.375 | |

| | | | |
|---|---|---|---|
| 71 | 1.32 | 1.32 | |
| 72 | 1.265 | 1.265 | |
| 73 | 1.21 | 1.21 | |
| 74 | 1.155 | 1.155 | |
| 75 | 1.1 | 1.1 | |
| 76 | 1.045 | 1.045 | |
| 77 | 0.99 | 0.99 | |
| 78 | 0.935 | 0.935 | |
| 79 | 0.88 | 0.88 | |
| 80 | 0.825 | 0.825 | 0. 83 |
| 81 | 0.77 | 0.77 | |
| 82 | 0.715 | 0.715 | |
| 83 | 0.66 | 0.66 | |
| 84 | 0.605 | 0.605 | |
| 85 | 0.55 | 0.55 | |
| 86 | 0.495 | 0.495 | |
| 87 | 0.44 | 0.44 | |
| 88 | 0.385 | 0.385 | |
| 89 | 0.33 | 0.33 | |
| 90 | 0.275 | 0.275 | |
| 91 | 0.22 | 0.22 | |
| 92 | 0.165 | 0.165 | |
| 93 | 0.11 | 0.11 | |
| 94 | 0.055 | 0.055 | |
| 95 | 0 | 0 | 0 |
| 96 | 0 | 0 | |
| 97 | 0 | 0 | |
| 98 | 0 | 0 | |
| 99 | 0 | 0 | |
| 100 | 0 | 0 | 0 |

2. Totally inaccurate ($\alpha$ =∞)

| $T_a$ | **Program Simulation** | **Calculation** | **STELLA** |
|---|---|---|---|
| 0 | 5.225 | 5.225 | 5. 22 |
| 1 | 25.18062241 | 25.335 | 24. 09067 |
| 2 | 25.58893909 | 25.3075 | |
| 3 | 25.73060317 | 25.28 | |
| 4 | 25.36369051 | 25.2525 | |
| 5 | 25.29815424 | 25.225 | |
| 6 | 25.7683305 | 25.1975 | |
| 7 | 25.33736107 | 25.17 | |
| 8 | 25.3708631 | 25.1425 | |
| 9 | 25.47874573 | 25.115 | |
| 10 | 24.71840485 | 25.0875 | |
| 11 | 25.03751627 | 25.06 | |
| 12 | 25.26017223 | 25.0325 | |
| 13 | 25.29301733 | 25.005 | |
| 14 | 24.94056021 | 24.9775 | |
| 15 | 25.95635802 | 24.95 | |
| 16 | 24.94446597 | 24.9225 | |
| 17 | 24.78594993 | 24.895 | |
| 18 | 24.71328344 | 24.8675 | |
| 19 | 24.66702656 | 24.84 | |
| 20 | 25.24286301 | 24.8125 | 23. 68773 |
| 21 | 25.44364913 | 24.785 | |
| 22 | 24.89574487 | 24.7575 | |
| 23 | 24.56284953 | 24.73 | |
| 24 | 25.05834006 | 24.7025 | |
| 25 | 24.74576585 | 24.675 | |
| 26 | 24.66637154 | 24.6475 | |
| 27 | 25.38719127 | 24.62 | |
| 28 | 24.70369495 | 24.5925 | |
| 29 | 24.90432398 | 24.565 | |
| 30 | 24.98778901 | 24.5375 | |
| 31 | 25.36360504 | 24.51 | |
| 32 | 24.70422207 | 24.4825 | |
| 33 | 24.19426858 | 24.455 | |
| 34 | 25.06588853 | 24.4275 | |
| 35 | 25.05483962 | 24.4 | |
| 36 | 24.19107329 | 24.3725 | |
| 37 | 25.01265883 | 24.345 | |
| 38 | 24.68031499 | 24.3175 | |

| | | | |
|---|---|---|---|
| 39 | 24.58418545 | 24.29 | |
| 40 | 24.74925198 | 24.2625 | 23.31527 |
| 41 | 24.59217671 | 24.235 | |
| 42 | 23.66285482 | 24.2075 | |
| 43 | 23.79860644 | 24.18 | |
| 44 | 24.58675245 | 24.1525 | |
| 45 | 24.16642761 | 24.125 | |
| 46 | 24.44530964 | 24.0975 | |
| 47 | 24.65701439 | 24.07 | |
| 48 | 24.42524237 | 24.0425 | |
| 49 | 24.34059418 | 24.015 | |
| 50 | 23.66966043 | 23.9875 | |
| 51 | 24.3578212 | 23.96 | |
| 52 | 24.35344667 | 23.9325 | |
| 53 | 24.28733296 | 23.905 | |
| 54 | 23.7727248 | 23.8775 | |
| 55 | 23.82943512 | 23.85 | |
| 56 | 23.58764596 | 23.8225 | |
| 57 | 23.9627997 | 23.795 | |
| 58 | 24.03165629 | 23.7675 | |
| 59 | 23.35873624 | 23.74 | |
| 60 | 23.59832179 | 23.7125 | 22.97473 |
| 61 | 24.25694034 | 23.685 | |
| 62 | 23.04954928 | 23.6575 | |
| 63 | 23.53859641 | 23.63 | |
| 64 | 24.03728867 | 23.6025 | |
| 65 | 23.81954056 | 23.575 | |
| 66 | 22.66146508 | 23.5475 | |
| 67 | 23.14068886 | 23.52 | |
| 68 | 23.29182628 | 23.4925 | |
| 69 | 22.89772498 | 23.465 | |
| 70 | 23.97210495 | 23.4375 | |
| 71 | 22.69546222 | 23.41 | |
| 72 | 23.1513152 | 23.3825 | |
| 73 | 23.42987575 | 23.355 | |
| 74 | 23.69467894 | 23.3275 | |
| 75 | 23.51181563 | 23.3 | |
| 76 | 23.05676607 | 23.2725 | |
| 77 | 23.09966361 | 23.245 | |
| 78 | 23.12682779 | 23.2175 | |
| 79 | 23.23182925 | 23.19 | |
| 80 | 23.19933538 | 23.1625 | 22.67253 |
| 81 | 23.36465748 | 23.135 | |

| | | | |
|---|---|---|---|
| 82 | 22.81851225 | 23.1075 | |
| 83 | 23.17444498 | 23.08 | |
| 84 | 23.25192583 | 23.0525 | |
| 85 | 23.28451006 | 23.025 | |
| 86 | 22.60515135 | 22.9975 | |
| 87 | 22.36944691 | 22.97 | |
| 88 | 22.95900334 | 22.9425 | |
| 89 | 22.55880969 | 22.915 | |
| 90 | 21.93765485 | 22.8875 | |
| 91 | 23.26268794 | 22.86 | |
| 92 | 22.4497635 | 22.8325 | |
| 93 | 23.03725102 | 22.805 | |
| 94 | 22.97896856 | 22.7775 | |
| 95 | 22.79611564 | 22.75 | |
| 96 | 23.27499538 | 22.75 | |
| 97 | 22.41528126 | 22.75 | |
| 98 | 22.52852756 | 22.75 | |
| 99 | 22.27568703 | 22.75 | |
| 100 | 22.35613139 | 22.75 | 22.465 |

3. Totally patience ($\beta$=0 or $\gamma$=T)

| $T_a$ | Program Simulation | Calculation | STELLA |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0.049639598 | 0.050675975 | |
| 2 | 0.10112912 | 0.101239069 | |
| 3 | 0.152926918 | 0.151689533 | |
| 4 | 0.200361695 | 0.202027617 | |
| 5 | 0.249844076 | 0.252253573 | |
| 6 | 0.306388471 | 0.30236765 | |
| 7 | 0.350550677 | 0.352370096 | |
| 8 | 0.401549306 | 0.402261162 | |
| 9 | 0.454478561 | 0.452041094 | |
| 10 | 0.48753413 | 0.501710141 | |
| 11 | 0.545230873 | 0.551268549 | |
| 12 | 0.600369935 | 0.600716565 | |
| 13 | 0.651730817 | 0.650054435 | |
| 14 | 0.690649794 | 0.699282404 | |
| 15 | 0.775353697 | 0.748400717 | |
| 16 | 0.790121718 | 0.797409618 | |
| 17 | 0.833964146 | 0.846309351 | |
| 18 | 0.88137883 | 0.895100159 | |
| 19 | 0.927875071 | 0.943782285 | |
| 20 | 1.00347823 | 0.992355971 | 0. 979467 |
| 21 | 1.062532961 | 1.040821458 | |
| 22 | 1.087053063 | 1.089178988 | |
| 23 | 1.119120805 | 1.1374288 | |
| 24 | 1.194306264 | 1.185571135 | |
| 25 | 1.229233853 | 1.233606232 | |
| 26 | 1.272190574 | 1.281534331 | |
| 27 | 1.366140106 | 1.329355669 | |
| 28 | 1.373432879 | 1.377070484 | |
| 29 | 1.435977937 | 1.424679013 | |
| 30 | 1.492337828 | 1.472181494 | |
| 31 | 1.568225962 | 1.519578162 | |
| 32 | 1.572321486 | 1.566869254 | |
| 33 | 1.586260265 | 1.614055004 | |
| 34 | 1.698350077 | 1.661135647 | |
| 35 | 1.74880591 | 1.708111417 | |
| 36 | 1.731281977 | 1.754982548 | |
| 37 | 1.847172506 | 1.801749272 | |
| 38 | 1.868215063 | 1.848411823 | |

| | | | |
|---|---|---|---|
| 39 | 1.909709595 | 1.894970433 | |
| 40 | 1.975474546 | 1.941425332 | 1. 918267 |
| 41 | 2.010717639 | 1.987776753 | |
| 42 | 1.974711385 | 2.034024925 | |
| 43 | 2.036687227 | 2.080170078 | |
| 44 | 2.15966536 | 2.126212443 | |
| 45 | 2.16735606 | 2.172152247 | |
| 46 | 2.245543346 | 2.21798972 | |
| 47 | 2.313597824 | 2.263725089 | |
| 48 | 2.340061025 | 2.309358581 | |
| 49 | 2.381396057 | 2.354890425 | |
| 50 | 2.359666542 | 2.400320846 | |
| 51 | 2.481412872 | 2.445650069 | |
| 52 | 2.530293046 | 2.490878321 | |
| 53 | 2.57234943 | 2.536005827 | |
| 54 | 2.562438452 | 2.58103281 | |
| 55 | 2.616038044 | 2.625959495 | |
| 56 | 2.638079194 | 2.670786105 | |
| 57 | 2.730062175 | 2.715512863 | |
| 58 | 2.78703714 | 2.760139992 | |
| 59 | 2.752252847 | 2.804667713 | |
| 60 | 2.829303915 | 2.849096248 | 2. 817533 |
| 61 | 2.960716517 | 2.893425817 | |
| 62 | 2.853491663 | 2.937656642 | |
| 63 | 2.964566159 | 2.981788942 | |
| 64 | 3.079009019 | 3.025822936 | |
| 65 | 3.099129461 | 3.069758844 | |
| 66 | 2.987224574 | 3.113596884 | |
| 67 | 3.100509907 | 3.157337275 | |
| 68 | 3.169403665 | 3.200980233 | |
| 69 | 3.160711968 | 3.244525975 | |
| 70 | 3.363024778 | 3.287974719 | |
| 71 | 3.223452681 | 3.33132668 | |
| 72 | 3.337153347 | 3.374582074 | |
| 73 | 3.426649902 | 3.417741115 | |
| 74 | 3.51466325 | 3.46080402 | |
| 75 | 3.533886218 | 3.503771001 | |
| 76 | 3.510782408 | 3.546642272 | |
| 77 | 3.56454052 | 3.589418047 | |
| 78 | 3.615753599 | 3.632098539 | |
| 79 | 3.679981458 | 3.674683958 | |
| 80 | 3.722156811 | 3.717174519 | 3. 670733 |
| 81 | 3.796562165 | 3.75957043 | |

| | | | |
|---|---|---|---|
| 82 | 3.752563238 | 3.801871905 | |
| 83 | 3.858807714 | 3.844079152 | |
| 84 | 3.919693878 | 3.886192382 | |
| 85 | 3.972208397 | 3.928211804 | |
| 86 | 3.901319504 | 3.970137627 | |
| 87 | 3.905623417 | 4.011970059 | |
| 88 | 4.056212899 | 4.053709309 | |
| 89 | 4.030965707 | 4.095355585 | |
| 90 | 3.964087024 | 4.136909092 | |
| 91 | 4.252252719 | 4.178370039 | |
| 92 | 4.148702941 | 4.219738631 | |
| 93 | 4.304495453 | 4.261015073 | |
| 94 | 4.340298822 | 4.302199572 | |
| 95 | 4.35209645 | 4.343292331 | |
| 96 | 4.485468672 | 4.384293556 | |
| 97 | 4.360095823 | 4.42520345 | |
| 98 | 4.422545257 | 4.466022216 | |
| 99 | 4.412789178 | 4.506750058 | |
| 100 | 4.468658692 | 4.547387179 | 4.493133 |

4. Totally impatience ($\beta$=1 and $\gamma$=0)

| $T_a$ | Program Simulation | Calculation | STELLA |
|---|---|---|---|
| 0 | 22 | 22 | 22 |
| 1 | 22.03763882 | 22.03842464 | |
| 2 | 22.07668032 | 22.07676369 | |
| 3 | 22.11595558 | 22.11501734 | |
| 4 | 22.1519226 | 22.15318578 | |
| 5 | 22.18944221 | 22.19126919 | |
| 6 | 22.23231653 | 22.22926778 | |
| 7 | 22.26580216 | 22.26718172 | |
| 8 | 22.30447145 | 22.30501121 | |
| 9 | 22.34460462 | 22.34275643 | |
| 10 | 22.36966874 | 22.38041758 | |
| 11 | 22.41341682 | 22.41799483 | |
| 12 | 22.45522555 | 22.45548838 | |
| 13 | 22.49416952 | 22.49289842 | |
| 14 | 22.52367951 | 22.53022512 | |
| 15 | 22.58790555 | 22.56746868 | |
| 16 | 22.59910328 | 22.60462927 | |
| 17 | 22.63234644 | 22.64170709 | |
| 18 | 22.66829823 | 22.67870232 | |
| 19 | 22.70355363 | 22.71561514 | |
| 20 | 22.7608791 | 22.75244574 | 22. 71413 |
| 21 | 22.80565686 | 22.78919429 | |
| 22 | 22.82424903 | 22.82586099 | |
| 23 | 22.84856413 | 22.86244601 | |
| 24 | 22.90557288 | 22.89894954 | |
| 25 | 22.93205644 | 22.93537176 | |
| 26 | 22.96462802 | 22.97171284 | |
| 27 | 23.03586448 | 23.00797298 | |
| 28 | 23.04139416 | 23.04415234 | |
| 29 | 23.08881844 | 23.08025112 | |
| 30 | 23.13155286 | 23.11626948 | |
| 31 | 23.18909441 | 23.15220762 | |
| 32 | 23.19219981 | 23.1880657 | |
| 33 | 23.20276877 | 23.2238439 | |
| 34 | 23.28775995 | 23.25954241 | |
| 35 | 23.32601767 | 23.2951614 | |
| 36 | 23.31273029 | 23.33070105 | |
| 37 | 23.40060333 | 23.36616154 | |
| 38 | 23.41655867 | 23.40154303 | |

| | | | |
|---|---|---|---|
| 39 | 23.44802156 | 23.43684571 | |
| 40 | 23.49788729 | 23.47206976 | 23.399 |
| 41 | 23.52461008 | 23.50721534 | |
| 42 | 23.49730863 | 23.54228264 | |
| 43 | 23.5443013 | 23.57727182 | |
| 44 | 23.63754846 | 23.61218306 | |
| 45 | 23.64337987 | 23.64701654 | |
| 46 | 23.70266473 | 23.68177242 | |
| 47 | 23.75426648 | 23.71645089 | |
| 48 | 23.77433199 | 23.75105211 | |
| 49 | 23.80567393 | 23.78557626 | |
| 50 | 23.78919771 | 23.8200235 | |
| 51 | 23.88151086 | 23.85439401 | |
| 52 | 23.91857385 | 23.88868796 | |
| 53 | 23.95046275 | 23.92290552 | |
| 54 | 23.94294784 | 23.95704686 | |
| 55 | 23.98358929 | 23.99111214 | |
| 56 | 24.00030181 | 24.02510155 | |
| 57 | 24.07004714 | 24.05901525 | |
| 58 | 24.11324794 | 24.0928534 | |
| 59 | 24.08687304 | 24.12661618 | |
| 60 | 24.14529637 | 24.16030375 | 24.05433 |
| 61 | 24.2449389 | 24.19391628 | |
| 62 | 24.16363654 | 24.22745394 | |
| 63 | 24.24785786 | 24.26091689 | |
| 64 | 24.33463321 | 24.2943053 | |
| 65 | 24.34988937 | 24.32761934 | |
| 66 | 24.26503841 | 24.36085918 | |
| 67 | 24.35093608 | 24.39402497 | |
| 68 | 24.40317421 | 24.42711688 | |
| 69 | 24.3965838 | 24.46013508 | |
| 70 | 24.54998582 | 24.49307973 | |
| 71 | 24.44415643 | 24.525951 | |
| 72 | 24.53036902 | 24.55874904 | |
| 73 | 24.59822905 | 24.59147403 | |
| 74 | 24.66496444 | 24.62412612 | |
| 75 | 24.6795401 | 24.65670548 | |
| 76 | 24.66202183 | 24.68921227 | |
| 77 | 24.70278347 | 24.72164665 | |
| 78 | 24.74161537 | 24.75400878 | |
| 79 | 24.79031561 | 24.78629883 | |
| 80 | 24.82229473 | 24.81851694 | 24.67667 |
| 81 | 24.87871197 | 24.85066329 | |

| | | | |
|---|---|---|---|
| 82 | 24.84535015 | 24.88273804 | |
| 83 | 24.92590915 | 24.91474133 | |
| 84 | 24.97207558 | 24.94667334 | |
| 85 | 25.01189428 | 24.97853422 | |
| 86 | 24.95814336 | 25.01032413 | |
| 87 | 24.96140677 | 25.04204323 | |
| 88 | 25.07559 | 25.07369167 | |
| 89 | 25.05644653 | 25.10526962 | |
| 90 | 25.00573631 | 25.13677722 | |
| 91 | 25.22423558 | 25.16821464 | |
| 92 | 25.14571981 | 25.19958204 | |
| 93 | 25.2638482 | 25.23087956 | |
| 94 | 25.29099581 | 25.26210737 | |
| 95 | 25.29994126 | 25.29326561 | |
| 96 | 25.40106965 | 25.32435445 | |
| 97 | 25.30600672 | 25.35537404 | |
| 98 | 25.35335849 | 25.38632454 | |
| 99 | 25.34596103 | 25.41720609 | 25.2468 |
| 100 | 4.468658692 | 4.547387179 | 4.493133 |

## Appendix - E. Calculation Results in MTB Parameter ($\alpha$) Analysis

$\alpha$ changing from 0.004 to 0.001 with a step of 0.0005 (the given conditions in section 5.3), the losses at different $T_a$ are:

| $T_a$ | $\alpha$=0.004 | $\alpha$=0.0035 | $\alpha$=0.003 | $\alpha$=0.0025 | $\alpha$=0.002 | $\alpha$=0.0015 | $\alpha$=0.001 |
|---|---|---|---|---|---|---|---|
| 0 | 7.98 | 7.98 | 7.98 | 7.98 | 7.98 | 7.98 | 7.98 |
| 1 | 8.013764 | 7.99907 | 7.984367 | 7.969658 | 7.954941 | 7.940217 | 7.925485 |
| 2 | 8.047059 | 8.017779 | 7.98847 | 7.959132 | 7.929764 | 7.900367 | 7.870941 |
| 3 | 8.079884 | 8.036129 | 7.992309 | 7.948422 | 7.90447 | 7.860452 | 7.816367 |
| 4 | 8.112244 | 8.054122 | 7.995884 | 7.93753 | 7.879059 | 7.82047 | 7.761764 |
| 5 | 8.144139 | 8.071759 | 7.999198 | 7.926455 | 7.85353 | 7.780422 | 7.707132 |
| 6 | 8.175572 | 8.089041 | 8.00225 | 7.915198 | 7.827884 | 7.740309 | 7.65247 |
| 7 | 8.206543 | 8.105968 | 8.005041 | 7.903759 | 7.802122 | 7.700129 | 7.597779 |
| 8 | 8.237056 | 8.122543 | 8.007572 | 7.892139 | 7.776244 | 7.659884 | 7.543059 |
| 9 | 8.267111 | 8.138767 | 8.009843 | 7.880339 | 7.75025 | 7.619574 | 7.488309 |
| 10 | 8.296712 | 8.15464 | 8.011857 | 7.868358 | 7.724139 | 7.579198 | 7.43353 |
| 11 | 8.325858 | 8.170165 | 8.013612 | 7.856197 | 7.697913 | 7.538756 | 7.378722 |
| 12 | 8.354553 | 8.185341 | 8.015111 | 7.843857 | 7.671572 | 7.49825 | 7.323884 |
| 13 | 8.382798 | 8.200172 | 8.016354 | 7.831338 | 7.645115 | 7.457678 | 7.269018 |
| 14 | 8.410595 | 8.214657 | 8.017341 | 7.81864 | 7.618543 | 7.417041 | 7.214122 |
| 15 | 8.437946 | 8.228798 | 8.018074 | 7.805765 | 7.591857 | 7.376339 | 7.159198 |
| 16 | 8.464853 | 8.242595 | 8.018553 | 7.792712 | 7.565056 | 7.335572 | 7.104244 |
| 17 | 8.491316 | 8.256052 | 8.018779 | 7.779481 | 7.538141 | 7.29474 | 7.049261 |
| 18 | 8.517339 | 8.269168 | 8.018753 | 7.766074 | 7.511111 | 7.253843 | 6.99425 |
| 19 | 8.542922 | 8.281944 | 8.018475 | 7.752491 | 7.483968 | 7.212882 | 6.939209 |
| 20 | 8.568068 | 8.294382 | 8.017946 | 7.738732 | 7.456712 | 7.171857 | 6.884139 |
| 21 | 8.592778 | 8.306484 | 8.017168 | 7.724798 | 7.429341 | 7.130767 | 6.829041 |
| 22 | 8.617054 | 8.318249 | 8.01614 | 7.710688 | 7.401858 | 7.089612 | 6.773913 |
| 23 | 8.640898 | 8.32968 | 8.014863 | 7.696404 | 7.374262 | 7.048394 | 6.718757 |
| 24 | 8.664312 | 8.340778 | 8.013339 | 7.681946 | 7.346553 | 7.007111 | 6.663572 |
| 25 | 8.687296 | 8.351543 | 8.011567 | 7.667315 | 7.318732 | 6.965765 | 6.608358 |
| 26 | 8.709854 | 8.361978 | 8.009549 | 7.65251 | 7.290798 | 6.924354 | 6.553115 |
| 27 | 8.731986 | 8.372082 | 8.007286 | 7.637532 | 7.262753 | 6.88288 | 6.497843 |
| 28 | 8.753694 | 8.381857 | 8.004778 | 7.622382 | 7.234595 | 6.841341 | 6.442543 |
| 29 | 8.774981 | 8.391305 | 8.002026 | 7.60706 | 7.206327 | 6.79974 | 6.387214 |
| 30 | 8.795847 | 8.400427 | 7.99903 | 7.591567 | 7.177946 | 6.758074 | 6.331857 |
| 31 | 8.816295 | 8.409223 | 7.995792 | 7.575903 | 7.149455 | 6.716345 | 6.276471 |
| 32 | 8.836325 | 8.417694 | 7.992312 | 7.560068 | 7.120853 | 6.674553 | 6.221056 |
| 33 | 8.855941 | 8.425843 | 7.98859 | 7.544063 | 7.09214 | 6.632698 | 6.165612 |
| 34 | 8.875142 | 8.43367 | 7.984628 | 7.527888 | 7.063316 | 6.590779 | 6.110141 |
| 35 | 8.893932 | 8.441176 | 7.980427 | 7.511543 | 7.034382 | 6.548798 | 6.05464 |
| 36 | 8.912311 | 8.448362 | 7.975986 | 7.49503 | 7.005339 | 6.506753 | 5.999111 |
| 37 | 8.930282 | 8.45523 | 7.971307 | 7.478348 | 6.976185 | 6.464645 | 5.943554 |
| 38 | 8.947846 | 8.46178 | 7.96639 | 7.461498 | 6.946922 | 6.422475 | 5.887968 |
| 39 | 8.965004 | 8.468014 | 7.961237 | 7.444481 | 6.917549 | 6.380242 | 5.832354 |
| 40 | 8.981758 | 8.473932 | 7.955847 | 7.427296 | 6.888068 | 6.337946 | 5.776712 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 41 | 8.99811 | 8.479536 | 7.950222 | 7.409945 | 6.858477 | 6.295588 | 5.721041 |
| 42 | 9.014062 | 8.484828 | 7.944362 | 7.392427 | 6.828778 | 6.253168 | 5.665341 |
| 43 | 9.029614 | 8.489807 | 7.938268 | 7.374743 | 6.79897 | 6.210685 | 5.609614 |
| 44 | 9.044769 | 8.494475 | 7.931941 | 7.356893 | 6.769054 | 6.16814 | 5.553858 |
| 45 | 9.059529 | 8.498834 | 7.925381 | 7.338878 | 6.73903 | 6.125532 | 5.498074 |
| 46 | 9.073894 | 8.502884 | 7.918589 | 7.320699 | 6.708898 | 6.082863 | 5.442262 |
| 47 | 9.087866 | 8.506626 | 7.911565 | 7.302355 | 6.678659 | 6.040132 | 5.386422 |
| 48 | 9.101447 | 8.510062 | 7.904311 | 7.283847 | 6.648312 | 5.997339 | 5.330553 |
| 49 | 9.114639 | 8.513192 | 7.896828 | 7.265176 | 6.617857 | 5.954484 | 5.274657 |
| 50 | 9.127443 | 8.516018 | 7.889115 | 7.246341 | 6.587296 | 5.911567 | 5.218732 |
| 51 | 9.13986 | 8.51854 | 7.881173 | 7.227344 | 6.556628 | 5.868589 | 5.162779 |
| 52 | 9.151892 | 8.52076 | 7.873004 | 7.208185 | 6.525854 | 5.825549 | 5.106798 |
| 53 | 9.163541 | 8.522679 | 7.864607 | 7.188863 | 6.494973 | 5.782448 | 5.05079 |
| 54 | 9.174809 | 8.524298 | 7.855984 | 7.169381 | 6.463986 | 5.739286 | 4.994753 |
| 55 | 9.185695 | 8.525618 | 7.847136 | 7.149737 | 6.432893 | 5.696063 | 4.938688 |
| 56 | 9.196204 | 8.526639 | 7.838062 | 7.129932 | 6.401694 | 5.652778 | 4.882595 |
| 57 | 9.206334 | 8.527363 | 7.828764 | 7.109967 | 6.37039 | 5.609432 | 4.826475 |
| 58 | 9.216089 | 8.527792 | 7.819242 | 7.089842 | 6.338981 | 5.566026 | 4.770327 |
| 59 | 9.22547 | 8.527925 | 7.809496 | 7.069558 | 6.307466 | 5.522558 | 4.71415 |
| 60 | 9.258083 | 8.527765 | 7.799529 | 7.049115 | 6.275847 | 5.47903 | 4.657946 |
| 61 | 9.330701 | 8.527311 | 7.789339 | 7.028513 | 6.244123 | 5.435441 | 4.601714 |
| 62 | 9.402862 | 8.526566 | 7.778929 | 7.007752 | 6.212295 | 5.391792 | 4.545455 |
| 63 | 9.474567 | 8.566706 | 7.768298 | 6.986834 | 6.180362 | 5.348082 | 4.489168 |
| 64 | 9.545818 | 8.626998 | 7.757447 | 6.965758 | 6.148325 | 5.304312 | 4.432853 |
| 65 | 9.616617 | 8.686932 | 7.746378 | 6.944525 | 6.116185 | 5.260481 | 4.37651 |
| 66 | 9.686966 | 8.746509 | 7.7745 | 6.923136 | 6.083941 | 5.21659 | 4.32014 |
| 67 | 9.756866 | 8.805732 | 7.822196 | 6.90159 | 6.051593 | 5.172639 | 4.263742 |
| 68 | 9.826319 | 8.864602 | 7.869623 | 6.879888 | 6.019142 | 5.128628 | 4.207316 |
| 69 | 9.895328 | 8.923119 | 7.916783 | 6.875122 | 5.986589 | 5.084557 | 4.150863 |
| 70 | 9.963893 | 8.981284 | 7.963675 | 6.909819 | 5.953932 | 5.040427 | 4.094382 |
| 71 | 10.03202 | 9.0391 | 8.010301 | 6.944324 | 5.921173 | 4.996236 | 4.037874 |
| 72 | 10.0997 | 9.096567 | 8.056661 | 6.978637 | 5.888311 | 4.951986 | 3.981339 |
| 73 | 10.16695 | 9.153687 | 8.102757 | 7.01276 | 5.882244 | 4.907676 | 3.924776 |
| 74 | 10.23376 | 9.21046 | 8.148589 | 7.046694 | 5.903264 | 4.863307 | 3.868185 |
| 75 | 10.30013 | 9.266888 | 8.194158 | 7.080437 | 5.924159 | 4.818878 | 3.811567 |
| 76 | 10.36608 | 9.322972 | 8.239465 | 7.113992 | 5.944928 | 4.77439 | 3.754922 |
| 77 | 10.43159 | 9.378713 | 8.284509 | 7.147357 | 5.965571 | 4.737399 | 3.698249 |
| 78 | 10.49667 | 9.434113 | 8.329294 | 7.180535 | 5.98609 | 4.74414 | 3.641549 |
| 79 | 10.56133 | 9.489173 | 8.373818 | 7.213525 | 6.006483 | 4.750809 | 3.584822 |
| 80 | 10.62556 | 9.543893 | 8.418083 | 7.246328 | 6.026752 | 4.757404 | 3.528068 |
| 81 | 10.68937 | 9.598276 | 8.46209 | 7.278943 | 6.046896 | 4.763927 | 3.471286 |
| 82 | 10.75275 | 9.652322 | 8.505839 | 7.311373 | 6.066916 | 4.770376 | 3.419574 |
| 83 | 10.81571 | 9.706032 | 8.549331 | 7.343617 | 6.086812 | 4.776753 | 3.411183 |
| 84 | 10.87826 | 9.759408 | 8.592567 | 7.375675 | 6.106585 | 4.783058 | 3.402759 |
| 85 | 10.94038 | 9.812451 | 8.635548 | 7.407548 | 6.126234 | 4.78929 | 3.394302 |
| 86 | 11.00209 | 9.865161 | 8.678274 | 7.439237 | 6.14576 | 4.79545 | 3.385811 |
| 87 | 11.06339 | 9.917541 | 8.720746 | 7.470742 | 6.165163 | 4.801538 | 3.377286 |
| 88 | 11.12428 | 9.969591 | 8.762966 | 7.502064 | 6.184444 | 4.807554 | 3.368728 |
| 89 | 11.18475 | 10.02131 | 8.804933 | 7.533202 | 6.203602 | 4.813497 | 3.360137 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 90 | 11.24481 | 10.07271 | 8.846648 | 7.564158 | 6.222637 | 4.819369 | 3.351512 |
| 91 | 11.30447 | 10.12377 | 8.888113 | 7.594932 | 6.241551 | 4.825169 | 3.342854 |
| 92 | 11.36372 | 10.17452 | 8.929328 | 7.625524 | 6.260343 | 4.830898 | 3.334162 |
| 93 | 11.42257 | 10.22493 | 8.970294 | 7.655934 | 6.279014 | 4.836555 | 3.325437 |
| 94 | 11.48101 | 10.27503 | 9.011011 | 7.686164 | 6.297563 | 4.84214 | 3.316679 |
| 95 | 11.53906 | 10.3248 | 9.05148 | 7.716213 | 6.315992 | 4.847654 | 3.307888 |
| 96 | 11.6387 | 10.41626 | 9.133702 | 7.788083 | 6.376299 | 4.895097 | 3.341063 |
| 97 | 11.73795 | 10.50739 | 9.215678 | 7.859773 | 6.436486 | 4.942469 | 3.374206 |
| 98 | 11.8368 | 10.59821 | 9.297408 | 7.931284 | 6.496553 | 4.98977 | 3.407315 |
| 99 | 11.93526 | 10.6887 | 9.378894 | 8.002617 | 6.5565 | 5.037 | 3.440391 |
| 100 | 12.03332 | 10.77888 | 9.460135 | 8.073771 | 6.616328 | 5.084159 | 3.473434 |

## Appendix - F.  Calculation Results in MTB Parameter ($\beta$) Analysis

$\beta$ changing from 0.002 to 0.012 with 0.002 increment (the given conditions in section 5.3), the losses at different $T_a$ are:

| $T_a$ | ***β=0.002*** | ***β=0.004*** | ***β=0.006*** | ***β=0.008*** | ***β=0.01*** | ***β=0.012*** |
|---|---|---|---|---|---|---|
| 0 | 2.66 | 5.32 | 7.98 | 10.64 | 13.3 | 14 |
| 1 | 2.705658 | 5.337658 | 7.969658 | 10.60166 | 13.23366 | 14.07366 |
| 2 | 2.751132 | 5.355132 | 7.959132 | 10.56313 | 13.16713 | 14.14713 |
| 3 | 2.796422 | 5.372422 | 7.948422 | 10.52442 | 13.10042 | 14.22042 |
| 4 | 2.84153 | 5.38953 | 7.93753 | 10.48553 | 13.03353 | 14.29353 |
| 5 | 2.886455 | 5.406455 | 7.926455 | 10.44645 | 12.96645 | 14.36645 |
| 6 | 2.931198 | 5.423198 | 7.915198 | 10.4072 | 12.8992 | 14.4392 |
| 7 | 2.975759 | 5.439759 | 7.903759 | 10.36776 | 12.83176 | 14.51176 |
| 8 | 3.020139 | 5.456139 | 7.892139 | 10.32814 | 12.76414 | 14.58414 |
| 9 | 3.064339 | 5.472339 | 7.880339 | 10.28834 | 12.69634 | 14.65634 |
| 10 | 3.108358 | 5.488358 | 7.868358 | 10.24836 | 12.62836 | 14.72836 |
| 11 | 3.152197 | 5.504197 | 7.856197 | 10.2082 | 12.5602 | 14.8002 |
| 12 | 3.195857 | 5.519857 | 7.843857 | 10.16786 | 12.49186 | 14.81586 |
| 13 | 3.239338 | 5.535338 | 7.831338 | 10.12734 | 12.42334 | 14.71934 |
| 14 | 3.28264 | 5.55064 | 7.81864 | 10.08664 | 12.35464 | 14.62264 |
| 15 | 3.325765 | 5.565765 | 7.805765 | 10.04576 | 12.28576 | 14.52576 |
| 16 | 3.368712 | 5.580712 | 7.792712 | 10.00471 | 12.21671 | 14.42871 |
| 17 | 3.411481 | 5.595481 | 7.779481 | 9.963481 | 12.14748 | 14.33148 |
| 18 | 3.454074 | 5.610074 | 7.766074 | 9.922074 | 12.07807 | 14.23407 |
| 19 | 3.496491 | 5.624491 | 7.752491 | 9.880491 | 12.00849 | 14.13649 |
| 20 | 3.538732 | 5.638732 | 7.738732 | 9.838732 | 11.93873 | 14.03873 |
| 21 | 3.580798 | 5.652798 | 7.724798 | 9.796798 | 11.8688 | 13.9408 |
| 22 | 3.622688 | 5.666688 | 7.710688 | 9.754688 | 11.79869 | 13.84269 |
| 23 | 3.664404 | 5.680404 | 7.696404 | 9.712404 | 11.7284 | 13.7444 |
| 24 | 3.705946 | 5.693946 | 7.681946 | 9.669946 | 11.65795 | 13.64595 |
| 25 | 3.747315 | 5.707315 | 7.667315 | 9.627315 | 11.58731 | 13.54731 |
| 26 | 3.78851 | 5.72051 | 7.65251 | 9.58451 | 11.51651 | 13.44851 |
| 27 | 3.829532 | 5.733532 | 7.637532 | 9.541532 | 11.44553 | 13.34953 |
| 28 | 3.870382 | 5.746382 | 7.622382 | 9.498382 | 11.37438 | 13.25038 |
| 29 | 3.91106 | 5.75906 | 7.60706 | 9.45506 | 11.30306 | 13.15106 |
| 30 | 3.951567 | 5.771567 | 7.591567 | 9.411567 | 11.23157 | 13.05157 |
| 31 | 3.991903 | 5.783903 | 7.575903 | 9.367903 | 11.1599 | 12.9519 |
| 32 | 4.032068 | 5.796068 | 7.560068 | 9.324068 | 11.08807 | 12.85207 |
| 33 | 4.072063 | 5.808063 | 7.544063 | 9.280063 | 11.01606 | 12.75206 |
| 34 | 4.111888 | 5.819888 | 7.527888 | 9.235888 | 10.94389 | 12.65189 |
| 35 | 4.151543 | 5.831543 | 7.511543 | 9.191543 | 10.87154 | 12.55154 |
| 36 | 4.19103 | 5.84303 | 7.49503 | 9.14703 | 10.79903 | 12.45103 |
| 37 | 4.230348 | 5.854348 | 7.478348 | 9.102348 | 10.72635 | 12.35035 |
| 38 | 4.269498 | 5.865498 | 7.461498 | 9.057498 | 10.6535 | 12.2495 |
| 39 | 4.308481 | 5.876481 | 7.444481 | 9.012481 | 10.58048 | 12.14848 |
| 40 | 4.347296 | 5.887296 | 7.427296 | 8.967296 | 10.5073 | 12.0473 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 41 | 4.385945 | 5.897945 | 7.409945 | 8.921945 | 10.43394 | 11.94594 |
| 42 | 4.424427 | 5.908427 | 7.392427 | 8.876427 | 10.36043 | 11.84443 |
| 43 | 4.462743 | 5.918743 | 7.374743 | 8.830743 | 10.28674 | 11.74274 |
| 44 | 4.516054 | 5.928893 | 7.356893 | 8.784893 | 10.21289 | 11.64089 |
| 45 | 4.583697 | 5.938878 | 7.338878 | 8.738878 | 10.13888 | 11.53888 |
| 46 | 4.651136 | 5.948699 | 7.320699 | 8.692699 | 10.0647 | 11.4367 |
| 47 | 4.718371 | 5.958355 | 7.302355 | 8.646355 | 9.990355 | 11.33435 |
| 48 | 4.785404 | 5.967847 | 7.283847 | 8.599847 | 9.915847 | 11.23185 |
| 49 | 4.852234 | 5.977176 | 7.265176 | 8.553176 | 9.841176 | 11.12918 |
| 50 | 4.918863 | 5.986341 | 7.246341 | 8.506341 | 9.766341 | 11.02634 |
| 51 | 4.98529 | 5.995344 | 7.227344 | 8.459344 | 9.691344 | 10.92334 |
| 52 | 5.051517 | 6.004185 | 7.208185 | 8.412185 | 9.616185 | 10.82018 |
| 53 | 5.117543 | 6.012863 | 7.188863 | 8.364863 | 9.540863 | 10.71686 |
| 54 | 5.183369 | 6.021381 | 7.169381 | 8.317381 | 9.465381 | 10.61338 |
| 55 | 5.248996 | 6.029737 | 7.149737 | 8.269737 | 9.389737 | 10.50974 |
| 56 | 5.314424 | 6.037932 | 7.129932 | 8.221932 | 9.313932 | 10.40593 |
| 57 | 5.379654 | 6.045967 | 7.109967 | 8.173967 | 9.237967 | 10.30197 |
| 58 | 5.444686 | 6.053842 | 7.089842 | 8.125842 | 9.161842 | 10.19784 |
| 59 | 5.509521 | 6.061558 | 7.069558 | 8.077558 | 9.085558 | 10.09356 |
| 60 | 5.574159 | 6.069115 | 7.049115 | 8.029115 | 9.009115 | 9.989115 |
| 61 | 5.6386 | 6.1146 | 7.028513 | 7.980513 | 8.932513 | 9.884513 |
| 62 | 5.702846 | 6.164846 | 7.007752 | 7.931752 | 8.855752 | 9.779752 |
| 63 | 5.766896 | 6.214896 | 6.986834 | 7.882834 | 8.778834 | 9.674834 |
| 64 | 5.830752 | 6.264752 | 6.965758 | 7.833758 | 8.701758 | 9.569758 |
| 65 | 5.894413 | 6.314413 | 6.944525 | 7.784525 | 8.624525 | 9.464525 |
| 66 | 5.95788 | 6.36388 | 6.923136 | 7.735136 | 8.547136 | 9.359136 |
| 67 | 6.021153 | 6.413153 | 6.90159 | 7.68559 | 8.46959 | 9.25359 |
| 68 | 6.084234 | 6.462234 | 6.879888 | 7.635888 | 8.391888 | 9.147888 |
| 69 | 6.147122 | 6.511122 | 6.875122 | 7.58603 | 8.31403 | 9.04203 |
| 70 | 6.209819 | 6.559819 | 6.909819 | 7.536018 | 8.236018 | 8.936018 |
| 71 | 6.272324 | 6.608324 | 6.944324 | 7.485851 | 8.157851 | 8.829851 |
| 72 | 6.334637 | 6.656637 | 6.978637 | 7.435529 | 8.079529 | 8.723529 |
| 73 | 6.39676 | 6.70476 | 7.01276 | 7.385053 | 8.001053 | 8.617053 |
| 74 | 6.458694 | 6.752694 | 7.046694 | 7.340694 | 7.922424 | 8.510424 |
| 75 | 6.520437 | 6.800437 | 7.080437 | 7.360437 | 7.843641 | 8.403641 |
| 76 | 6.581992 | 6.847992 | 7.113992 | 7.379992 | 7.764706 | 8.296706 |
| 77 | 6.643357 | 6.895357 | 7.147357 | 7.399357 | 7.685618 | 8.189618 |
| 78 | 6.704535 | 6.942535 | 7.180535 | 7.418535 | 7.656535 | 8.082378 |
| 79 | 6.765525 | 6.989525 | 7.213525 | 7.437525 | 7.661525 | 7.974986 |
| 80 | 6.826328 | 7.036328 | 7.246328 | 7.456328 | 7.666328 | 7.876328 |
| 81 | 6.886943 | 7.082943 | 7.278943 | 7.474943 | 7.670943 | 7.866943 |
| 82 | 6.947373 | 7.129373 | 7.311373 | 7.493373 | 7.675373 | 7.857373 |
| 83 | 7.007617 | 7.175617 | 7.343617 | 7.511617 | 7.679617 | 7.847617 |
| 84 | 7.067675 | 7.221675 | 7.375675 | 7.529675 | 7.683675 | 7.837675 |
| 85 | 7.127548 | 7.267548 | 7.407548 | 7.547548 | 7.687548 | 7.827548 |
| 86 | 7.187237 | 7.313237 | 7.439237 | 7.565237 | 7.691237 | 7.817237 |
| 87 | 7.246742 | 7.358742 | 7.470742 | 7.582742 | 7.694742 | 7.806742 |
| 88 | 7.306064 | 7.404064 | 7.502064 | 7.600064 | 7.698064 | 7.796064 |
| 89 | 7.365202 | 7.449202 | 7.533202 | 7.617202 | 7.701202 | 7.785202 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 90 | 7.424158 | 7.494158 | 7.564158 | 7.634158 | 7.704158 | 7.774158 |
| 91 | 7.482932 | 7.538932 | 7.594932 | 7.650932 | 7.706932 | 7.762932 |
| 92 | 7.541524 | 7.583524 | 7.625524 | 7.667524 | 7.709524 | 7.751524 |
| 93 | 7.599934 | 7.627934 | 7.655934 | 7.683934 | 7.711934 | 7.739934 |
| 94 | 7.658164 | 7.672164 | 7.686164 | 7.700164 | 7.714164 | 7.728164 |
| 95 | 7.716213 | 7.716213 | 7.716213 | 7.716213 | 7.716213 | 7.716213 |
| 96 | 7.788083 | 7.788083 | 7.788083 | 7.788083 | 7.788083 | 7.788083 |
| 97 | 7.859773 | 7.859773 | 7.859773 | 7.859773 | 7.859773 | 7.859773 |
| 98 | 7.931284 | 7.931284 | 7.931284 | 7.931284 | 7.931284 | 7.931284 |
| 99 | 8.002617 | 8.002617 | 8.002617 | 8.002617 | 8.002617 | 8.002617 |
| 100 | 8.073771 | 8.073771 | 8.073771 | 8.073771 | 8.073771 | 8.073771 |

# Appendix - G. Calculation Results in MTB Parameter ($\gamma$) Analysis

$\gamma$ is in range of 5 to 35 with a step of 10 (the given conditions in section 5.3), the losses at different $T_a$ are:

| $T_a$ | $\gamma$=5 | $\gamma$=15 | $\gamma$=25 | $\gamma$=35 |
|---|---|---|---|---|
| 0 | 7.98 | 7.14 | 6.3 | 5.46 |
| 1 | 7.969658 | 7.129658 | 6.289658 | 5.449658 |
| 2 | 7.959132 | 7.119132 | 6.279132 | 5.439132 |
| 3 | 7.948422 | 7.108422 | 6.268422 | 5.428422 |
| 4 | 7.93753 | 7.09753 | 6.25753 | 5.41753 |
| 5 | 7.926455 | 7.086455 | 6.246455 | 5.406455 |
| 6 | 7.915198 | 7.075198 | 6.235198 | 5.395198 |
| 7 | 7.903759 | 7.063759 | 6.223759 | 5.383759 |
| 8 | 7.892139 | 7.052139 | 6.212139 | 5.372139 |
| 9 | 7.880339 | 7.040339 | 6.200339 | 5.360339 |
| 10 | 7.868358 | 7.028358 | 6.188358 | 5.348358 |
| 11 | 7.856197 | 7.016197 | 6.176197 | 5.336197 |
| 12 | 7.843857 | 7.003857 | 6.163857 | 5.323857 |
| 13 | 7.831338 | 6.991338 | 6.151338 | 5.311338 |
| 14 | 7.81864 | 6.97864 | 6.13864 | 5.29864 |
| 15 | 7.805765 | 6.965765 | 6.125765 | 5.285765 |
| 16 | 7.792712 | 6.952712 | 6.112712 | 5.272712 |
| 17 | 7.779481 | 6.939481 | 6.099481 | 5.259481 |
| 18 | 7.766074 | 6.926074 | 6.086074 | 5.246074 |
| 19 | 7.752491 | 6.912491 | 6.072491 | 5.232491 |
| 20 | 7.738732 | 6.898732 | 6.058732 | 5.218732 |
| 21 | 7.724798 | 6.884798 | 6.044798 | 5.204798 |
| 22 | 7.710688 | 6.870688 | 6.030688 | 5.190688 |
| 23 | 7.696404 | 6.856404 | 6.016404 | 5.176404 |
| 24 | 7.681946 | 6.841946 | 6.001946 | 5.161946 |
| 25 | 7.667315 | 6.827315 | 5.987315 | 5.147315 |
| 26 | 7.65251 | 6.81251 | 5.97251 | 5.13251 |
| 27 | 7.637532 | 6.797532 | 5.957532 | 5.117532 |
| 28 | 7.622382 | 6.782382 | 5.942382 | 5.102382 |
| 29 | 7.60706 | 6.76706 | 5.92706 | 5.08706 |
| 30 | 7.591567 | 6.751567 | 5.911567 | 5.071567 |
| 31 | 7.575903 | 6.735903 | 5.895903 | 5.055903 |
| 32 | 7.560068 | 6.720068 | 5.880068 | 5.040068 |
| 33 | 7.544063 | 6.704063 | 5.864063 | 5.024063 |
| 34 | 7.527888 | 6.687888 | 5.847888 | 5.007888 |
| 35 | 7.511543 | 6.671543 | 5.831543 | 4.991543 |
| 36 | 7.49503 | 6.65503 | 5.81503 | 4.97503 |
| 37 | 7.478348 | 6.638348 | 5.798348 | 4.958348 |
| 38 | 7.461498 | 6.621498 | 5.781498 | 4.941498 |
| 39 | 7.444481 | 6.604481 | 5.764481 | 4.924481 |
| 40 | 7.427296 | 6.587296 | 5.747296 | 4.907296 |

| | | | | |
|---|---|---|---|---|
| 41 | 7.409945 | 6.569945 | 5.729945 | 4.889945 |
| 42 | 7.392427 | 6.552427 | 5.712427 | 4.872427 |
| 43 | 7.374743 | 6.534743 | 5.694743 | 4.854743 |
| 44 | 7.356893 | 6.516893 | 5.676893 | 4.836893 |
| 45 | 7.338878 | 6.498878 | 5.658878 | 4.818878 |
| 46 | 7.320699 | 6.480699 | 5.640699 | 4.800699 |
| 47 | 7.302355 | 6.462355 | 5.622355 | 4.802371 |
| 48 | 7.283847 | 6.443847 | 5.603847 | 4.841404 |
| 49 | 7.265176 | 6.425176 | 5.585176 | 4.880234 |
| 50 | 7.246341 | 6.406341 | 5.566341 | 4.918863 |
| 51 | 7.227344 | 6.387344 | 5.547344 | 4.95729 |
| 52 | 7.208185 | 6.368185 | 5.528185 | 4.995517 |
| 53 | 7.188863 | 6.348863 | 5.508863 | 5.033543 |
| 54 | 7.169381 | 6.329381 | 5.491369 | 5.071369 |
| 55 | 7.149737 | 6.309737 | 5.528996 | 5.108996 |
| 56 | 7.129932 | 6.289932 | 5.566424 | 5.146424 |
| 57 | 7.109967 | 6.269967 | 5.603654 | 5.183654 |
| 58 | 7.089842 | 6.249842 | 5.640686 | 5.220686 |
| 59 | 7.069558 | 6.229558 | 5.677521 | 5.257521 |
| 60 | 7.049115 | 6.209115 | 5.714159 | 5.294159 |
| 61 | 7.028513 | 6.188513 | 5.7506 | 5.3306 |
| 62 | 7.007752 | 6.206846 | 5.786846 | 5.366846 |
| 63 | 6.986834 | 6.242896 | 5.822896 | 5.402896 |
| 64 | 6.965758 | 6.278752 | 5.858752 | 5.438752 |
| 65 | 6.944525 | 6.314413 | 5.894413 | 5.474413 |
| 66 | 6.923136 | 6.34988 | 5.92988 | 5.55188 |
| 67 | 6.90159 | 6.385153 | 5.965153 | 5.629153 |
| 68 | 6.879888 | 6.420234 | 6.000234 | 5.706234 |
| 69 | 6.875122 | 6.455122 | 6.035122 | 5.783122 |
| 70 | 6.909819 | 6.489819 | 6.069819 | 5.859819 |
| 71 | 6.944324 | 6.524324 | 6.104324 | 5.936324 |
| 72 | 6.978637 | 6.558637 | 6.138637 | 6.012637 |
| 73 | 7.01276 | 6.59276 | 6.17276 | 6.08876 |
| 74 | 7.046694 | 6.626694 | 6.206694 | 6.164694 |
| 75 | 7.080437 | 6.660437 | 6.240437 | 6.240437 |
| 76 | 7.113992 | 6.693992 | 6.315992 | 6.315992 |
| 77 | 7.147357 | 6.727357 | 6.391357 | 6.391357 |
| 78 | 7.180535 | 6.760535 | 6.466535 | 6.466535 |
| 79 | 7.213525 | 6.793525 | 6.541525 | 6.541525 |
| 80 | 7.246328 | 6.826328 | 6.616328 | 6.616328 |
| 81 | 7.278943 | 6.858943 | 6.690943 | 6.690943 |
| 82 | 7.311373 | 6.891373 | 6.765373 | 6.765373 |
| 83 | 7.343617 | 6.923617 | 6.839617 | 6.839617 |
| 84 | 7.375675 | 6.955675 | 6.913675 | 6.913675 |
| 85 | 7.407548 | 6.987548 | 6.987548 | 6.987548 |
| 86 | 7.439237 | 7.061237 | 7.061237 | 7.061237 |
| 87 | 7.470742 | 7.134742 | 7.134742 | 7.134742 |
| 88 | 7.502064 | 7.208064 | 7.208064 | 7.208064 |
| 89 | 7.533202 | 7.281202 | 7.281202 | 7.281202 |

| | | | | |
|---|---|---|---|---|
| 90 | 7.564158 | 7.354158 | 7.354158 | 7.354158 |
| 91 | 7.594932 | 7.426932 | 7.426932 | 7.426932 |
| 92 | 7.625524 | 7.499524 | 7.499524 | 7.499524 |
| 93 | 7.655934 | 7.571934 | 7.571934 | 7.571934 |
| 94 | 7.686164 | 7.644164 | 7.644164 | 7.644164 |
| 95 | 7.716213 | 7.716213 | 7.716213 | 7.716213 |
| 96 | 7.788083 | 7.788083 | 7.788083 | 7.788083 |
| 97 | 7.859773 | 7.859773 | 7.859773 | 7.859773 |
| 98 | 7.931284 | 7.931284 | 7.931284 | 7.931284 |
| 99 | 8.002617 | 8.002617 | 8.002617 | 8.002617 |
| 100 | 8.073771 | 8.073771 | 8.073771 | 8.073771 |