# Graduation Project
# Fortimedix Surgical

## Defining a system for a self-controlled supply of a production line


| | |
|---|---|
| Author: | L.H. (Loek) Vroomen |
| Student no.: | 0803942 |
| Professor: | Prof. Dr. Ir. Adan |
| Supervisor: | Dr. Ir. Lefeber |
| Document no.: | 2017.073 |

Department of Mechanical Engineering
Manufacturing Networks


August 14, 2017

# Abstract

Fortimedix Surgical has developed a manufacturing line to manufacture symphonX$^{\mathrm{TM}}$ instruments. This manufacturing line consists of manufacturing processes and assembly processes. Each of these processes is supplied with components that are either bought or manufactured in house. The assembly line is currently not operating on its maximum capacity. Within a year this production line is going to operate on >95% of its maximum capacity to be able to accommodate the projected growth of the business. Therefore, it is important to regulate the inventory and inventory supply of the production line to reduce inventory costs and to reduce stock outs during production. This research aims to design a matlab calculation tool which optimizes the optimal inventory policy. Using this calculation tool results in a self-controlled supply of the assembly line.

First an insight into the theory of the $(Q, s)$ policy and the $(R, S)$ policy is gathered. Detailed information on processes are investigated and collected to implement the $(Q, s)$ policy and the $(R, S)$ policy. Matlab scripts are developed to determine the optimal inventory policies. These scripts use an input file with parameter and is compiled into a matlab calculation tool. The calculation tool for the $(Q, s)$ policy calculates the optimal inventory costs, optimal reorder levels and order quantities. The calculation tool for the $(R, S)$ policy calculates the optimal inventory costs, optimal order levels and optimal review period. Both tools also calculate the average amount of full time employees per day which handle all orders. The calculation tools are expanded with bin size restrictions which implies that order quantities have a maximum limit. All parameters including the maximum bin sizes can be adapted in the input file and the optimal inventory policy is determined.

A simulation model of the complete assembly line is developed in Simevents. This simulation model is used to check whether the theoretical optimal inventory policies determined from the calculation tool are correct. This simulation model without supply of products can be used to determine lead times of instruments, utilizations of operators and utilizations of processes.

First a partial simulation model is established to simulate the supply of components using the $(Q, s)$ policy. Parameters derived from the $(Q, s)$ policy calculation tool are used as input parameters for the simulation model. The expected average inventory and amount of replenishment of the simulation model and calculation tool are equal which validates the $(Q, s)$ policy calculation tool. The next step is the implementation of the $(Q, s)$ policy into the complete assembly line simulation model. However, inserting an inventory policy for the supply of components results in a run time error. Therefore, the validation of the $(Q, s)$ policy on the complete simulation model is not succeeded neither the validation of the $(R, S)$ policy calculation tool.

Calculations showed that bin size restriction has a mayor influence on the total inventory costs. Not only do the costs increase, but also the amount of FTE to handle the orders increases significantly. Cost reduction can be realized using the inventory policy and the calculation tool.

# Contents

# Chapter 1

# Introduction

This chapter gives a brief introduction into the company Fortimedix. Which products are produced at Fortimedix and what is the motivation of this research.

## 1.1 Fortimedix

Fortimedix was founded in 1999 and is a global leader in the field of endovascular stents and an innovative product definition company, specialized in minimal invasive surgery. In 2009 Fortimedix started a subsidiary company named Fortimedix Surgical which is specialized in minimal invasive surgery. Fortimedix Surgical developed a Laparoscopic Surgical Instrument (LSI) named symphonX$^{TM}$, which launched on the market in October 2016.

## 1.2 Products

symphonX$^{TM}$ is the world's first single-port surgery solution compatible with a standard 15 mm laparoscopic trocar which means that a surgery can be performed through using only one single incision of a diameter of 15mm. Through this incision the symphonX$^{TM}$ surgical platform can be inserted. Figure 1.1 shows the symphonX$^{TM}$ surgical platform. As can be seen, the platform consists of the introducer which is a framework which can hold two instruments, a camera and a suction and irrigation device.

Figure 1.1: The complete symphonX$^{TM}$ product with two instruments

Currently, a total of six types of symphonX$^{TM}$ instruments are manufactured at Fortimedix Surgical. The different types are: Scissor, Grasper, Maryland, Clip applier, Hook-knife and a Suction irrigation instrument. Figure 1.1 shows two of the above mentioned instrument types, namely the Hook-knife instrument (upper instrument) and the Grasper instrument.

## 1.3 Department Motivation

Fortimedix Surgical has developed a manufacturing line to manufacture symphonX$^{TM}$ instruments. This manufacturing line consists of manufacturing processes and assembly processes. Each of these processes is supplied with components that are either bought or manufactured in house. The assembly line is currently not operating on its maximum capacity. Within a year this production line is going to operate on >95% of its maximum capacity to be able to accommodate the projected growth of the business. Therefore it is important to regulate the inventory and inventory supply of the production line to reduce inventory costs and to reduce stock outs during production.

This chapter provided a brief introduction into the company and the products produced. The motivation for this research is given. The next chapter describes the research objective and gives an insight into all processes at Fortimedix which are needed to produce an symphonX$^{TM}$ instrument.

# Chapter 2

# Research objective

The previous chapter gave a brief introduction into Fortimedix and its products. This chapter describes the research objective and gives an insight into all processes at Fortimedix which are needed to produce a symphonX$^{TM}$ instrument. Finally the research plan is explained.

## 2.1 Objective

All processes at Fortimedix have different process times, set up times and capacity restrictions. Therefore, it is important that the amount of inventory parts are chosen correctly to prevent processes from running out of supplies and to reduce inventory costs. Currently, the processes do not have an optimal system to supply all processes with the required components.

All manufactured components have a different process time and can be manufactured on various machines. The bin sizes of processes are chosen arbitrarily. Due to different process times and set-up times between component changes, it is important to choose the bin sizes of the manufactured components correctly to result in a maximum output of the process. Also to minimize immediate supply issues and redundant costs.

Due to the arbitrary choice of bin sizes and not optimal component supply, inventory costs are to large or processes run out of supplies causing processes to be disrupted and priorities change to address immediate supply issues. Therefore, the goal of this project is to determine the optimum inventory policy for the processes which need components.

The scope of this project is limited to the processes which need components at the assembly line in the clean room. More insight into the processes is given in the following sections.

## 2.2 Processes

The symphonX$^{TM}$ manufacturing line can be divided into two major parts, inside and outside the clean room. Firstly the assembly line which is located in the clean room. The process steps are given in Figure 2.1. Due to company secrecy, the process steps are fictional.

As explained in Section 1.2, there are six types of symphonX$^{TM}$ instruments. All of these instrument types can be produced with only three different types of product bodies. A product body is manufactured at the first station of the assembly line, see Figure 2.1. At this workstation a number of components are combined resulting in a product body. Depending on which parts are combined, different body types can be made. At process A1, only manufactured components derived from the manufactured component flow are needed. These components are manufactured outside the clean room. More detailed information about the manufactured components is given in the next sections.

The next assembly step is the process A2. During this process, the product bodies are merged with the buy-in components resulting in a functional product. All components needed at the

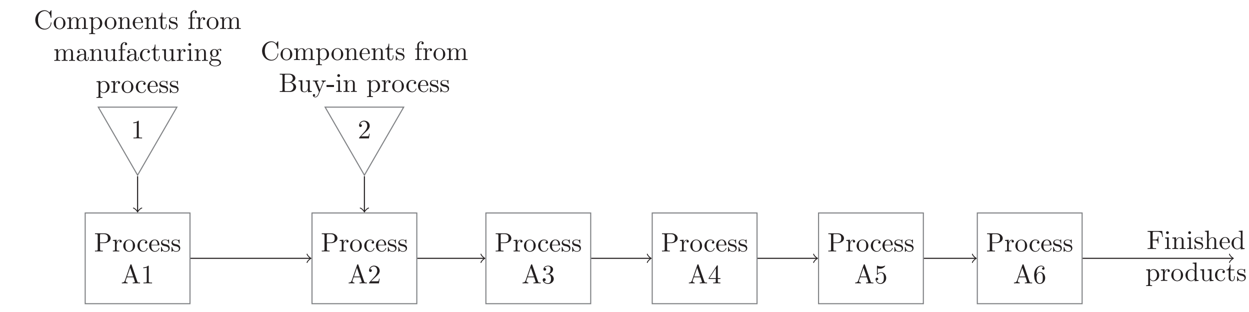


Figure 2.1: Assembly process flow in the clean room

process A2 are also prepared outside the clean room. The buy-in components derive from the buy-in component flow which is explained later on.

If the instruments are finished with process A2, they are tested at process A3. At this workstation the products are bent and checked whether they meet the bending requirements. Also, small tests are performed to check if the instruments meet all additional requirements. The passed instruments are moved to the next process A4 and A5. The final assembly step is process A6, after which the instruments are transported outside the clean room. During assembly all instruments are processed separately and piece wise. All used components of the instruments can be traced back to its origin.

The second part of the manufacturing line of Fortimedix provides the supply of various components for the final instruments. It is divided in the manufactured component flow and the buy-in component flow. The first step of the manufacturing flow is process M1 where raw materials enter the factory and are processed into components. These components are stored in a small stock keeping unit W1. Where after these components need some additional processing before they can be used in a final instrument. Figure 2.2 shows the process flow for manufactured components.

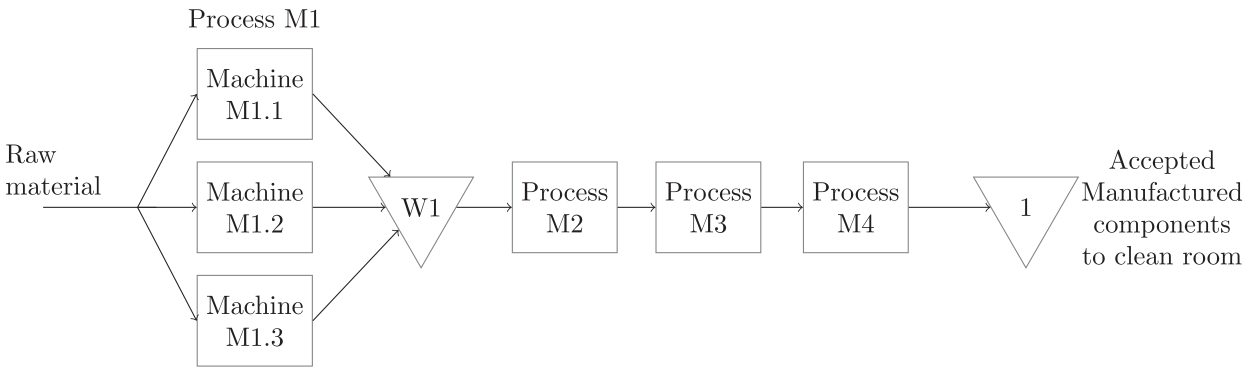


Figure 2.2: Process flow for manufactured components

The supply of buy-in components is also located outside the clean room. All buy-in components need to be processed before they can be used in the final instruments. Figure 2.3 shows the buy-in component flow. As can be seen, a partial amount of the buy-in components are processed at process B1 and some products can go directly to a small stock keeping unit W2. Where after all components proceed with process B2 and B3. Finally, the components are transferred to the clean room where they can be used in the assembly line.

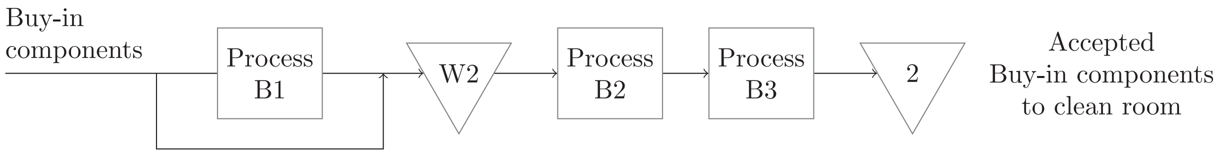


Figure 2.3: Buy in product flow

## 2.3 Research plan

The previous sections gave an insight into to research objective and into the processes used to produce a symphonX$^{TM}$ instrument. In this section the research plan is explained.

The first step of the research plan is to collect theory of the inventory policies. With this theory, it is possible to apply the inventory policies to the processes of Fortimedix. The next step is to make a matlab script to calculate the theoretical optimal inventory policies. Then a simulation of the assembly line is made to check whether these theoretical optimal inventory policies are correct. If so, the matlab script is validated and can be used as calculation tool for Fortimedix to determine the optimal inventory policies.

For this research additional information is required. What is the optimal inventory? Most research investigates the minimal cost for inventory in the production line to be the optimal inventory. However, this depends a lot on the service rate of the production line. For Fortimedix, the optimal inventory is a balance between cost of inventory and the service rate. The following topics are needed to determine the optimal inventory:

- Process times and yield of all assembly processes
- Required finished products per day/week/month
- Lead-time and yield for all component processes
- Production planning of process M1
- All costs per product (Order costs, holding cost)
- Storage boundaries for stock
- Safety margins of stock

Firstly, the symphonX$^{TM}$ assembly line layout is investigated to determine the process times for each workstation. The demand of finished goods needs to be determined, resulting in the demand of components for all workstations. The manufactured and buy-in component flow are monitored to determine the lead times of all components. Also the required costs per product are calculated. All volume constraints for inventory and bin sizes are investigated. Currently the safety stock margins are arbitrarily chosen, namely one month production volume. However this margin is probably not optimal. This research determines the optimal inventory levels for all components. The next chapter describes all the above listed information.

# Chapter 3

# Theoretical review

The previous chapter gave an insight into all processes at Fortimedix. Also the research objective and research plan is discussed. This chapter provides all theoretical information needed to legalise the research plan. Firstly, an insight into the theoretical view of inventory control is gathered. The two different inventory policies used in this research are discussed. All cost functions and variables per policy are given which are used in the next chapter to determine the optimal inventory costs per policy.

## 3.1 Input variables of inventory systems

As explained in Section 2.3, various parameters are needed to determine the optimal inventory policies. The first variable needed for inventory control is the demand of components. Demand can be constant or variable over time. The demand rate at Fortimedix is stochastic, but the average demand rate is assumed constant. This means that Fortimedix determines how many products are produced per day. However, due to the assembly line efficiency, the amount of start-ups per day can differ from the required amount of products per day.

Another important variable of inventory control is the lead time. Items which are ordered normally are not delivered instantaneously. The time between ordering components and the actual delivery is called lead time and is represented as $\tau$ . The lead time and demand are expressed in the same time unit.

In the inventory theory, there are two ways the check the inventory levels in time (review time). Namely continuous review, which means that the level of inventory on hand is always known, or periodic review, which means that at periodic time moments the inventory is known. For example, every week the inventory levels are checked.

Another important topic for inventory systems is excess demand. If demand is larger than the current inventory, the system can react in two different ways. Namely, excess demand can result in immediate loss of production or components can be back ordered with associated costs.

All above mentioned variables and topics to determine the optimal inventory policies are elaborated in the next sections.

## 3.2 Demand

A normal distribution is determined by two parameters: the mean $\mu$ and the variance $\sigma^2$. These can be estimated from a history of demand by the sample mean $\bar{D}$ and the sample variance $s^2$. Let $D_1, D_2, ..., D_n$ be $n$ past observations of demand. Then

$$\bar{D} = \frac{1}{n}\sum_{i=1}^{n} D_i, \qquad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(D_i - \bar{D})^2 \tag{3.1}$$

Demand is assumed normal distributed and there are two types of demand at Fortimedix. The amount of products that has to be produced per day, and the product types per day. In the near future, Fortimedix wants to produce all standard product types alternately. However, the suction irrigation device is produced separately in the available scheduled production time. Section 4.7 gives an insight in the forecast for the demand per product.

## 3.3 Costs

For Fortimedix an inventory optimization is described which must result in optimal inventory costs. Therefore, all associated cost for inventory optimization need to be researched. There are three types of cost. Ordering cost, holding costs and back ordering costs. All type of costs are explained in next subsections.

### 3.3.1 Holding Cost

All costs associated with holding inventory are known as the holding costs. Holding costs can be divided into several categories. A major portion of the holding costs are determined by the costs of capital of the inventory. Physical storage also determines the amount of holding costs because inventory needs to be handled and processed. In addition, costs associated with risk of inventory such as insurance, deterioration, breakage and obsolescence also need to be included with the holding cost. Holding costs can be expressed as a proportional ratio ($I$) to the costs of an item($c$).

$$h = I \cdot c. \tag{3.2}$$

The cost of an item consist of several costs. The purchase costs, the machine costs of producing one item and the operator costs per item. The standard 'rule of thumb' for this proportional ratio is 25% of the physical inventory value [1]. However, Fortimedix uses a proportional ratio of 7% per year. The holding costs for all components are calculated in Chapter 4.

### 3.3.2 Order Cost

In addition to holding costs, order costs do not depend on the current inventory. The order costs are dependent on the number of products ordered [2]. Normally the order costs consist of two parts. The fixed costs, denoted by $K$ and the variable costs, denoted by $c$. The fixed cost can be associated with the setup cost which are size independent. In contrast to the fixed costs, the variable costs are size dependent. The order cost of $x$ units can be defined as:

$$C(x) = \begin{cases} 0 & \text{if } x = 0, \\ K + cx & \text{if } x > 0. \end{cases} \tag{3.3}$$

When ordering from an external supplier, ordering costs are more expensive than internal ordering. Within the scope of this project only internal ordering is included. Costs related to ordering cost are costs to place an order, costs to transport an order and costs for handling an order. At Fortimedix these cost differ for both product flows; the product flow for buy-in products and the product flow for manufactured components. The order costs for all processes and components are gathered and calculated in Chapter 4.

### 3.3.3 Penalty Cost

If inventory on hand is insufficient to satisfy desired demand, the missing items need to be back-ordered. The associated costs are called penalty costs. Penalty costs can be interpreted in two ways: items can be back-ordered and delivered in next delivery or production rate is lower (loss of sales). If products are back ordered, processes could be interrupted which cause inefficiency. If production rate decreases due to back orders, penalty costs includes loss of profit. The penalty costs are denoted by symbol $\pi$.

## 3.4 EOQ Model

A basic inventory model is the economic order quantity (EOQ) model. The EOQ model describes the ratio of holding costs and order costs for a given Q. The EOQ model defines the optimal Q which minimizes the total ordering costs and holding costs. The EOQ model is used as basis for more complex analyses. An EOQ model has a few assumptions, the demand rate is assumed constant, back ordering is not allowed and the order lead time is zero. The costs that are included in the EOQ model are:

1. Setup costs, $K$, per order.
2. Order costs, $c$, per unit .
3. Holding cost, $h$, per unit per time unit .

The average annual cost is divided in the annual setup costs, order costs and holding cost and is given by [2]

$$G(Q) = \frac{K\lambda}{Q} + \lambda c + \frac{hQ}{2}. \tag{3.4}$$

To find the optimal order quantity ($Q$) the derivative of the annual costs needs to be equal to 0.

$$G'(Q) = 0, \qquad Q^* = \sqrt{\frac{2K\lambda}{h}}. \tag{3.5}$$

An example of an EOQ model is given. For this example the parameters of one of the buy-in components (part 401218) are used . Ordering a batch of this part costs 2.16 euro per order, $K = 2.16$. The daily demand ($\lambda$) is equal to 25.15 products with a corresponding holding cost($h$) of 0.000148095 euro per part. Order costs per unit are not included in this example because this part is ordered per batch. Using these parameters, the optimal order quantity can be calculated using (3.5):

$$Q^* = \sqrt{\frac{2 * 2.16 * 25.15}{0.000148095}} = 856.53. \tag{3.6}$$

The optimal order quantity for this component is 857 pieces. This is visualized in Figure 3.1. As can be seen, the optimal ratio between order cost and holding cost is established with an order quantity of 857. The holding costs are calculated using the average amount of pieces on inventory ($\frac{Q}{2}$) multiplied with the costs of holding one piece on stock and is displayed with the blue line in the graph. The setup costs are calculated using the amount of orders ($\frac{\lambda}{Q}$) multiplied with the costs of one order, displayed with the red line in the graph. The summation of these two costs results in the average day cost displayed with the yellow line.

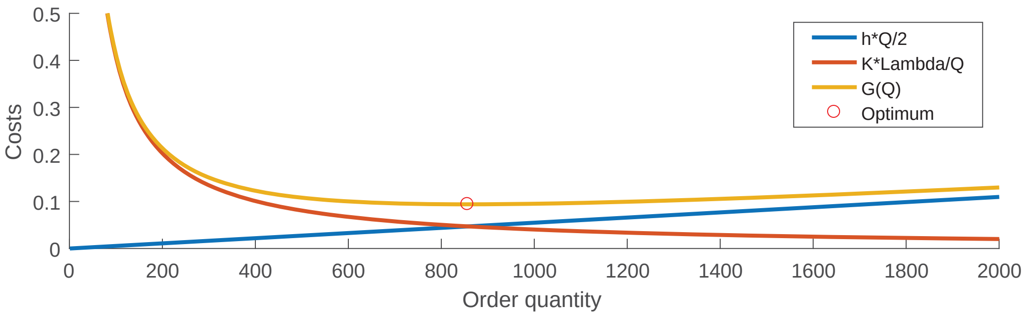


Figure 3.1: EOQ graph of buy-in product 401218

## 3.5 Inventory policies

### 3.5.1 $(Q, s)$ Policy

As mentioned in previous section, there are two types of review times in inventory systems. Firstly the continuous review policy, $(Q, s)$ policy, is explained. The $(Q, s)$ policy is also called the order quantity, reorder point system with a stochastic demand. A typical pattern of inventory progression determined by a $(Q, s)$ policy with non-deterministic demand is shown in Figure 3.2.

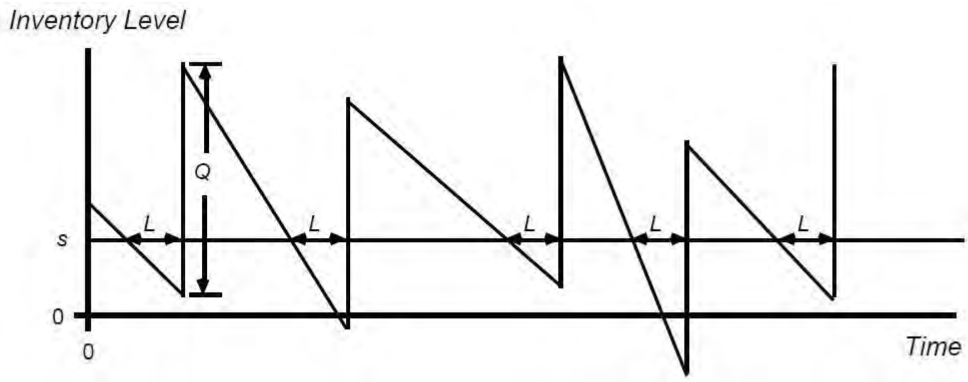


Figure 3.2: $(Q, s)$ Policy with non-deterministic demand

As can be seen in Figure 3.2, if the inventory level is below a certain reorder point, $s$, an order of lot size $Q$ is placed. The inventory replenishment occurs after lead time $L$ has passed. In this example, the decrease of inventory is displayed as a straight line, however demand ($\mu$) is normally not constant but stochastic. For the implementation of the policy, the values $Q$ and $s$ are required.

As Jensen and Bard [3] explain, a general cost model for the $(Q, s)$ policy can be developed. The assumptions made regarding costs can affect the optimal solution of the policy. The model only includes randomness of the demand, other uncertainties are excluded which makes the model an approximation of reality. This model uses the same notations as mentioned above and in the EOQ model in Section 3.4.
All variables need to have the same time dimensions, therefore the demand rate per time unit is displayed as $a$. The average inventory level can be determined if continuous demand is assumed using;

$$\text{Average inventory level} = \tfrac{Q}{2} + s - \mu. \tag{3.7}$$

Holding costs per time unit can be calculated using the average inventory level multiplied with the holding costs ($h$)

$$\text{Expected Holding Cost per unit time} = h(\tfrac{Q}{2} + s - \mu). \tag{3.8}$$

If back orders are allowed, the time between orders can be assumed random with an average value of lot size ($Q$) divided by demand rate per time unit ($a$). The expected costs per time unit for replenishment can be calculated using the setup costs ($K$) for each order and the order time. Resulting in;

$$\text{Expected Replenishment Cost per unit time} = \tfrac{Ka}{Q}. \tag{3.9}$$

As Jensen and Bard [3] further state, if the lead time of an order is relatively smaller than the order cycle time and an $(Q, s)$ policy is assumed, the shortage costs per cycle only depend on the reorder point ($s$). The shortage costs per cycle is a function of $s$ and defined by $C_s$. The shortage costs per time unit can be calculated by dividing $C_s$ with the cycle time;

$$\text{Expected Shortage Cost per unit time} = \tfrac{a}{Q} C_s \tag{3.10}$$

A combination of these expected costs establishes the general model for the $(Q, s)$ policy.

$$EC(Q, s) = h(\frac{Q}{2} + s - \mu) + \frac{Ka}{Q} + \frac{a}{Q} C_s \tag{3.11}$$

As can be seen in (3.11), the two variables of the cost function are $s$ and $Q$. Taking the partial derivative of the cost function, the minimal costs of the policy is established. The partial derivative with respect to $Q$ is given in (3.12).

$$\frac{\delta EC}{\delta Q} = \frac{h}{2} - \frac{a(K + Cs)}{Q^2} = 0, \qquad Q^* = \sqrt{\frac{2a(K + C_s)}{h}}. \tag{3.12}$$

The partial derivative with respect to $s$ is given in (3.13).

$$\frac{\delta EC}{\delta s} = h + \frac{a}{Q} \frac{\delta C_s}{\delta s} = 0, \qquad \frac{\delta C_s}{\delta s} = -\frac{hQ}{a}. \tag{3.13}$$

However, the partial derivative with respect to $s$ still relates to $s$ in the shortage costs. The costs of shortage can be considered in different cases. For example, a fixed cost if a stock out occurs. A shortage occurs if the demand during lead time is greater than $s$. This probability is defined as $P_s$,

$$P_s = P\{x > s\} = \int_s^\infty f(x)dx = 1 - F(s). \tag{3.14}$$

Then it is possible to express the expected shortage cost per cycle. If shortage occurs the penalty costs need to be included. The expected shortage cost is defined as,

$$C_s = \pi_1 P\{x > s\} = \pi_1 \int_s^\infty f(x)dx. \tag{3.15}$$

Then take the partial derivative of (3.15)

$$\frac{\delta C_s}{\delta s} = -\pi_1 f(s) \tag{3.16}$$

The combination of (3.13) and (3.16) results in the optimum reorder point $s$.

$$\frac{\delta C_s}{\delta s} = -\pi_1 f(s) = -\frac{hQ}{a} \tag{3.17}$$

$$\text{or, } f(s^*) = \tfrac{hQ}{\pi_1 a} \tag{3.18}$$

$$\text{and } C_s = \pi_1[1 - F(s^*)]. \tag{3.19}$$

If the parameters of (3.18) are known and the demand is assumed to have a normal distribution, it is possible to calculate the probability density function (p.d.f.) of the Standard Normal distribution. Using the Standard Normal tables the $z^*$ value is found and using (3.20) the optimum reorder point $s^*$ can be calculated.

$$s^* = \mu + (z^*)\sigma. \tag{3.20}$$

Finally, it is possible to define the service level of the inventory policy. The service level is the probability that the inventory will not be depleted during one order cycle. The probability of stock out is known (3.27) and therefore, the service level can be defined as:

$$\text{Service level} = 1 - P_s = F(s) \tag{3.21}$$

### 3.5.2 $(R, S)$ Policy

Besides continuous review of inventory it is also possible to only check inventory periodically. The time length between two review points is indicated by $R$. The inventory is checked each review moment and replenished until the desired order level $S$. This policy is called the $(R, S)$ policy and is illustrated in Figure 3.3

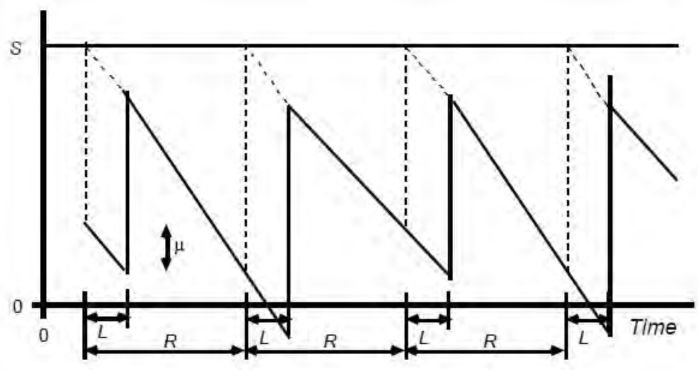


Figure 3.3: (R,S) Policy with non-deterministic demand

As can be seen in Figure 3.3, all intervals between review points have the same length. Because the demand differs at all times, the amount of replenishment is not always equal. This policy also replenishes inventory if the lead time has elapsed. The solid line displays the inventory on hand, and the dotted line displays the inventory position which includes the on hand inventory and the ordered inventory.

As Jensen and Bard [4] explain, the $(Q, s)$ and $(R, S)$ policy are similar to another. The biggest difference is that the $(Q, s)$ policy accounts for the shortage during lead time $L$, whereas the $(R, S)$ policy accounts for the shortage during lead time and interval $L + R$. Because the interval $L + R$ is much larger than only the lead time, the $(R, S)$ policy is more influenced by variability than the $(Q, s)$ policy. However, an advantage of the $(R, S)$ policy is that continuous review is not required.

Similar to the $(Q, s)$ policy, demand can be defined as random variable $X$ but now for interval $L + R$. The p.d.f and c.d.f (cumulative distribution function) displayed as $f_P(x)$ and $F_P(x)$ respectively can be found in the Standard Normal tables. The mean and variance of the periodic demand during the interval $L + R$ is displayed as $\mu_P$ and $\sigma_P$

Equation 3.9 showed that the time between orders could be expressed as $\frac{Q}{a}$. However at the $(R, S)$ policy the time between orders is known as $R$, resulting in $\frac{Q}{a} = R$. Substituting this in (3.8), (3.9) ,(3.10) and (3.11) the several costs and expected shortage for the $(R, S)$ policy are established.

$$\text{Average inventory level} = \tfrac{aR}{2} + S - \mu_P. \tag{3.22}$$

$$\text{Expected Holding Cost per unit time} = h(\tfrac{aR}{2} + S - \mu_P.). \tag{3.23}$$

$$\text{Expected Replenishment Cost per unit time} = \tfrac{K}{R}. \tag{3.24}$$

$$\text{Expected Shortage Cost per unit time} = \tfrac{1}{R}C_s \tag{3.25}$$

A summation of these expected costs results in the general model for the $(R, S)$ policy,

$$EC(R, S) = h(\frac{aR}{2} + S - \mu_P) + \frac{K}{R} + \frac{1}{R}C_s \tag{3.26}$$

A shortage occurs if the demand during lead time and review period is greater than $S$. This probability is defined as $P_S$,

$$P_S = P\{x > S\} = \int_S^\infty f(x)dx = 1 - F(S). \tag{3.27}$$

Then it is possible to express the expected shortage cost per cycle. If shortage occurs, the penalty costs need to be included. The penalty costs for the $(R, S)$ policy are defined as a fixed cost, $\pi_2$, if stock out occurs. The expected shortage cost is defined as,

$$C_S = \pi_2 P\{x > S\} = \pi_2 \int_S^\infty f(x)dx = \pi_2[1 - F(S)] \tag{3.28}$$

### 3.5.3 Outline

This chapter gave an insight in the theoretical view of inventory control. The theoretical information needed to verify the research plan is given. The two different inventory policies used to verify the research plan are discussed. An insight into the cost functions and variables per policy are given. The next chapter provides detailed information on the process times of the assembly processes. The lead times of the manufactured component flow and buy-in component flow are determined, and all associated costs to determine the inventory policies are described in detail. Also the sales forecast is given to determine the optimal inventory policies for the forecast.

# Chapter 4

# Data Analysis

The previous chapter gave an insight in the theoretical view of inventory control. The theoretical information needed to verify the research plan was given. The two different inventory policies used to verify the research plan were discussed. An insight into the cost functions and variables per policy are given. This chapter provides detailed information on the process times of the assembly processes. The lead times of the manufactured component flow and buy-in component flow are determined. All associated costs to determine the inventory policies are described in detail. Finally the sales forecast is given to determine the optimal inventory policies for the forecast.

## 4.1 Process times and yield of all assembly processes

As explained in Section 2.2, all accepted components go to the assembly process in the clean room. Currently, the assembly processes that need components are process A1 and process A2. Therefore these processes are investigated in more detail.

### 4.1.1 Process A1

At process A1, components are combined and merged resulting in a product body. At this process, three types of bodies can be manufactured, namely the 'standard' body, the hook knife body and the suction irrigation body. The 'standard' body and the hook-knife body can be manufactured arbitrarily. To produce the suction irrigation device a time schedule has to be made. The technical manager decides which bodies are made each day by looking at the stock levels of finished goods. Process A1 consists of two workstations. The second workstation is considered a flexible workstation, which will only manufacture products when needed. Momentarily, the desired amount of daily production is 25 products, which means that the gross start up is 28 products.

All process times are timed manually and also documented by the operators. The manually timed data of process A1 is given in Table 4.1.

| Product type | Standard Body | Hook-knife Body | Suction Irrigation Body |
|---|---|---|---|
| Assembly time [min] | 11.75 | 11.75 | 7 |
| Standard deviation [min] | 1.38 | 1.38 | 1.67 |

Table 4.1: Manually timed data of process A1

A working day has 7,5 hours which means that without any interruptions, approximately $\frac{7.5*60}{11.75} = 38$ standard or hook-knife bodies can be produced at one workstation of process A1 per day. This is significantly more than the amount of desired products per day. Currently, one operator is available per day for process A1, this operator documents the products made per day with the corresponding time. From this data, it appears that the average process time is also approximately 12 minutes as indicated in Table 4.1. However, this data also shows that the average occupation time of process A1 is 333 minutes per day, which corresponds to approximately 28 products made per day.

### 4.1.2 Process A2

At the second process of the assembly line, the diversity of products is established. The Scissor, Grasper, Maryland and Clip applier are almost identical. The only difference is that these products differ in two components. The hook-knife also consist of a few different components. Because there are some differences in component merging, the process times of the standard body and hook-knife body are slightly different. In addition, the suction irrigation has a completely different instrument assembly. The assembly data of process A2 is given in Table 4.2.

| Product type | Standard Body | Hook-knife Body | Suction Irrigation Body |
|---|---|---|---|
| Assembly time [min] | 9.92 | 9 | 5 |
| Standard deviation [min] | 1.65 | 1.55 | N/A |

Table 4.2: Manually timed data of process A2

As can be seen in Table 4.2, the standard deviation of the assembly time of the suction irrigation is not known. This is because the suction irrigation device was not produced during the time of the project. The assembly time shown in the table is derived from one product produced.

## 4.2 Lead-time and yield for all component processes

### 4.2.1 Buy-in component preparation

Buy-in components are stored in the large warehouse. A small amount of these products is stored in a small warehouse. Therefore, operators can easily access these products to prepare for production. The clean room determines which components need to be prepared for production with aid of a kanban system. If an empty kanban box leaves the clean room, the next day that box needs to be filled with products again.

As explained in Section 2.2 and Figure 2.3, a partial amount of the buy-in components need to be pre-processed before they can be used in the final product. Pre-processing of these components is done once a week for four hours. During these four hours, the warehouse boxes are (re-)filled. Components are assumed available when needed.

An operator is needed at process B2 for 5 minutes where one bin is processed. Then the operator transports the bin with components to process B3 where the components are processed for 2 hours. Total preparation and finishing a bin takes approximately 4 minutes, more detailed information is given in section 3.3.2. Summarized the lead time for ordering buy-in components is $4 + 5 + 120 = 129$ minutes. For the policy calculations in next chapter, all components are assumed sufficiently available.

### 4.2.2 Manufacturing component flow

As shown in Section 2.2 and Figure 2.2 there are currently three machines available for process M1 at Fortimedix. Two machines of brand T and one machine of brand R. The T-machines can be divided in a short and long edition on which different parts can be made. Additionally the T-machines have a double product output which allows the T-machines to produce two products in one cycle run per part. The R-machine can only produce one product per cycle run. The T-machines are equipped with a component loader for automatic feed of raw materials. The T-machine-long can handle a maximum of 2x22 units of raw materials, the T-machine-short 2x10 units of raw materials. The R-machine can only handle one unit of raw materials. All processed components have other process times and are processed on other machines. Table 4.3 shows the components processed at process M1 with their process time and process machine.

After the raw materials are processed they are transferred to process M2. First, the components derived from process M1 are prepared to go to process M2. Process M2 takes approximately 15 minutes. Then an operator action is performed at process M2. Finally, the components are positioned in a bin and processed for 2 hours at process M3. All components have a maximum number which can be processed at one time at process M2. Therefore, the bin sizes of process M1 are adjusted to the restricted bin sizes of process M2. These bin sizes are also given in Table 4.3.

When the components are finished with process M2 and process M3, the components are transported bin by bin to process M4. At process M4, the components are processed piece wise. Currently 100% of the components are processed at process M4 due to the quality performance of process M1. The process times of process M4 per component are also shown in Table 4.3.

| Part Number | Process M1 machine time [min/run] | Type Machine Process M1 | Bin size | Process M4 process time [sec/pc] | Yield [%] |
|---|---|---|---|---|---|
| 401127 | 4.14 | TS | 60 | 74 | 95 |
| 401131 | 34.6 | TL | 30 | 111 | 78 |
| 401132 | 21.5 | TL | 22 | 123 | 90 |
| 401133 | 15.4 | TS | 30 | 104 | 75 |
| 401134 | 2.5 | TS | 50 | 35 | 89 |
| 401136 | 3.37 | TS | 50 | 40* | 84 |
| 401137 | 3.14 | R | 50 | 26 | 95* |
| 401145 | 30* | TL | 25 | 120* | 88* |
| 401146 | 30* | TL | 25 | 80* | 95* |
| 401222 | 3.5* | TS | 50 | 40* | 90* |

Table 4.3: Process M1 info per part number.

Currently, Fortimedix does not use a periodical production planning to produce components at process M1. Fortimedix is still developing and improving many products which leads to uncertain planability. However, to calculate the inventory policies and costs, a planning for process M1 has to be established. This periodic planning ensures that all components are made on time and a safety stock can be build. The products made per day per machine are listed in Table 4.4 and Table 4.5. These production numbers are calculated by means of loader capacity of the machines, setup times and process time per part. These schedules repeat every two weeks, Table 4.4 shows the component production schedule of the standard instruments, and Table 4.5 shows the component production of the Suction Irrigation instrument. Because the desired amount of Suction Irrigation devices is 1/11 part of total production, the planning shown in Table 4.5 is executed every eleventh production cycle. Which means that week 1 until 20 the standard components are produced and week 21 and 22 the Suction Irrigation components are produced. Afterwards the cycle repeats.

| Week 1 | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday/Sunday |
|---|---|---|---|---|---|---|
| Machine TS | Part 27 (290 pc.) | Part 27 (350 pc.) | Part 36 (160 pc.)<br>Part 34 (560 pc.) | Part 33 (150 pc.) | Part 33 (66 pc.) | Part 33 (100 pc.) |
| Machine TL | Part 31 (80 pc.) | Part 32 (115 pc.) | Part 32 (125 pc.) | Part 32 (125 pc.) | Samples | Part 31 (220 pc.) |
| Machine R | Part 37 (200 pc.) | Part 37 (200 pc.) | Part 37 (200 pc.) | Samples | Samples | Samples |

| Week 2 | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday/Sunday |
|---|---|---|---|---|---|---|
| Machine TS | Part 33 (160 pc.) | Part 33 (160 pc.) | Part 33 (160 pc.) | Samples | Samples | Part 33 (100 pc.) |
| Machine TL | Part 31 (80 pc.) | Part 31 (80 pc.) | Part 32 (115 pc.) | Part 32 (125 pc.) | Part 31 (22 pc.) | Part 31 (220 pc.) |
| Machine R | Samples | Samples | Samples | Samples | Samples | Samples |

Table 4.4: Periodic production planning for process M1 standard products

| Week 1 | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday/Sunday |
|---|---|---|---|---|---|---|
| Machine TS | Part 22 (880 pc.) | Samples | Samples | Samples | Samples | |
| Machine TL | Part 45 (88 pc.) | Part 45 (96 pc.) | Part 45 (96 pc.) | Part 45 (96 pc.) | Part 45 (32 pc.) | Part 45 (176 pc.) |
| Machine R | Part 45 (16 pc.) | Part 45 (16 pc.) | Part 45 (16 pc.) | Part 45 (16 pc.) | Part 45 (12 pc.) | Part 45 (4 pc.) |

| Week 1 | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday/Sunday |
|---|---|---|---|---|---|---|
| Machine TS | Samples | Samples | Samples | Samples | Samples | Samples |
| Machine TL | Part 46 (88 pc.) | Part 46 (96 pc.) | Part 46 (96 pc.) | Part 46 (96 pc.) | Part 46 (32 pc.) | Part 46 (220 pc.) |
| Machine R | Part 45 (16 pc.) | Part 45 (16 pc.) | Part 45 (16 pc.) | Part 45 (16 pc.) | Part 45 (12 pc.) | Part 45 (4 pc.) |

Table 4.5: Periodic production planning for process M1 Suction Irrigation products

## 4.3 Order costs

As explained in Section 3.3.2, there are two parts of order costs (fixed costs and variable costs). The actions to be taken per ordering one batch are the same, regardless of the number of buy-in components in the bin, until the maximum volume of a bin. If the order quantity is larger than the maximum volume of a bin, an extra bin is ordered and the costs are added. The order cost of the manufactured components can be determined because the actions taken for an order and process times are known. The fixed order cost per day include: pick up of empty bins in the clean room (10 min per day) and bring products to clean room (10 min per day).

As explained in Section 3.3.1, the costs of the products include all process costs, so only the additional actions related to processing batches are considered order costs. The variable order cost for placing an order for buy-in components include:

- Counting products which are required in a bin (2min per bin)
- Take components from process B3 and put them in the bin (1 min per bin)
- Provide bin with traceability information (1 min per bin) and place bin in transport car

Using (3.3) the variable order cost for the buy-in component flow can be calculated. As can be seen, the actions taken are all independent of the volume of the box however the maximum amount of products in a box is restricted. The action times for ordering one bin are added and result in 4 minutes of process time. Knowing that the hourly rates of an operator is equal to 32.4 euros, the variable order costs for ordering one bin is equal to 2.16 euro. The delivery and collection of bins to and from the clean room is only done once per day. The total process time is 20 minutes which results in a fixed ordering cost of 10.8 euro. The order costs formula is given in (4.1). The variable $x$ represents the amount of bin ordered, $K$ represents the fixed order costs and $c$ represents the variable order costs.

$$C(x) = \begin{cases} 0 & \text{if } x = 0, \\ K + cx & \text{if } x > 0. \end{cases} \qquad C(x) = \begin{cases} 0 & \text{if } x = 0, \\ 10.8 + 2.16\,x & \text{if } x > 0. \end{cases} \tag{4.1}$$

The next step is to calculate the variable order costs of the manufactured component flow. The fixed order costs of the manufactured components only depend on the transport to the clean room. These fixed costs are already incorporated in the fixed order costs of the buy-in components because both buy-in and manufactured components are transported together to the clean room.

The ordering costs of the manufactured components only include the addition actions which are not included in the process costs.

- Make batch card (1 min per batch)
- Finish batch, log on pc (1 min per batch)
- Transport finished batches to process M2 (5 min per batch)
- Additional actions process M2 (5 min per batch)
- Transport and finish batches to process M4 (7 min per batch)

The total sum of the action time per batch is 19 minutes. The hourly rate for an operator is 32.4 euro resulting in a variable ordering cost of 10.26 euro per batch.

## 4.4 Holding Costs

As mentioned in Section 3.3.1, the holding costs can be expressed as a proportional ratio to the costs of an item. For company secrecy, the given item costs are fictional. However these fictional costs can easily be changed in the calculation tool to calculate the optimal inventory policy. All costs per product are displayed in Table A1.1 in Appendix A1 and the holding costs are calculated based on a yearly holding percentage of 7%. Also the current amount of kanban bins with the current corresponding content are shown in the table. The table also shows which item is proceeded at the manufactured component flow and which components are processed at the buy-in component flow.

## 4.5 Lead time

The lead time per component can be calculated using the order time and the process time. For the buy-in component flow the order time, process B2 time and process B3 time is added. For the manufactured component flow the order time, process M2 time, process M3 time and process M4 is added. However the lead time of the manufactured components depends on the batch size because all manufactured components are processed piece-wise at process M2. The calculation of the lead times are given in Table 4.6. These lead times are used to calculate the optimal inventory costs. The variable $Q$ represents the batch size of the order.

| Part | Order time [min/batch] | Process B2/M2 time[min/batch] | Process B3/M3 time [min/batch] | Process M4 [sec/piece] | Leadtime [min/batch] |
|---|---|---|---|---|---|
| 401127 | 19 | 20 | 120 | 74 | 159+1.23$Q$ |
| 401131 | 19 | 20 | 120 | 111 | 159+1.85$Q$ |
| 401132 | 19 | 20 | 120 | 123 | 159+2.05$Q$ |
| 401133 | 19 | 20 | 120 | 104 | 159+1.73$Q$ |
| 401134 | 19 | 20 | 120 | 35 | 159+0.58$Q$ |
| 401136 | 19 | 20 | 120 | 40 | 159+0.66$Q$ |
| 401137 | 19 | 20 | 120 | 26 | 159+0.43$Q$ |
| 401145 | 19 | 20 | 120 | 120 | 159+2$Q$ |
| 401146 | 19 | 20 | 120 | 80 | 159+1.33$Q$ |
| 401222 | 19 | 20 | 120 | 40 | 159+0.66$Q$ |
| Buy-in Components | 4 | 5 | 120 | - | 129 |

Table 4.6: The composition of the leadtimes per component

## 4.6 Penalty Costs

As explained in Section 3.3.3, if on hand inventory can not satisfy the desired demand, penalty costs need to be included in the cost model. At Fortimedix, the penalty costs are based on the downtime of the production line. This is because the actions to be taken for a normal delivery do not differ from a penalty order. Penalty orders must be carried out urgently rather than carried out at a chosen time. For all components, the penalty costs are different, due to the fact that different processes could shut down according to the lacking component. For example, if a manufactured component needed at process A1 is not available, three operators run out of work during the order lead time of that component. One operator working at process A1, one operator working at process A2 and one operator working at process A3. These penalty costs can be calculated for all components using the lead time per product and the amount of operators which are shut down. Because the lead time depends on the batch size, also the penalty costs depend on the batch size. For example, the lead time of part 401127 is 159+1.23$Q$ minutes and $Q$ represents the batch size. If this part lacks, three people are affected. The penalty costs will result in 159+1.23$Q$ minutes $\times$ 3 operators $\times$ 32.4 euro/hour = 257.5+1.99$Q$ euro. All penalty costs are shown in Table 4.7.

| Part | Leadtime [min/batch] | Affected Staff [operators] | Penalty Costs [Euros/batch] |
|---|---|---|---|
| 401127 | $159+1.23Q$ | 3 | $257.6+1.99Q$ |
| 401131 | $159+1.85Q$ | 3 | $257.6+2.99Q$ |
| 401132 | $159+2.05Q$ | 3 | $257.6+3.32Q$ |
| 401133 | $159+1.73Q$ | 3 | $257.6+2.8Q$ |
| 401134 | $159+0.58Q$ | 3 | $257.6+0.94Q$ |
| 401136 | $159+0.66Q$ | 3 | $257.6+1.07Q$ |
| 401137 | $159+0.43Q$ | 2 | $171.7+0.46Q$ |
| 401145 | $159+2Q$ | 3 | $257.6+3.24Q$ |
| 401146 | $159+1.33Q$ | 3 | $257.6+2.15Q$ |
| 401222 | $159+0.66Q$ | 3 | $257.6+1.07Q$ |
| Buy-in Components | 129 | 2 | 139.32 |

Table 4.7: Penalty cost calculation

## 4.7 Sales forecast

Due to the fact that Fortimedix is in start up phase, the two fictional possible scenarios of ramp-up forecast are given in Table 4.8. The numbers shown in the table represent the amount of surgery procedures which will be performed. An estimation has been made that during one surgery all 5 standard products are used and one at two times the Suction Irrigation device. For example, the forecast for year 2018 is 2750 procedures per year. Which corresponds to a total of $2750\times5=13750$ mixed standard products per year and $2750\times0.5=1375$ suction irrigation devices per year. The forecast shown in Table 4.8 can be used to calculate the optimal inventory cost per forecast scenario and inventory policy.

| Scenario #every type product | 2017 | 2018 | 2019 |
|---|---|---|---|
| Sales scenario A | 780 | 2750 | 5500 |
| Sales scenario B | 780 | 2750 | 4125 |

Table 4.8: Ramp-up forecast

Currently, Fortimedix does not sell products yet. Only clinical trails consume products during surgery resulting in an increase of warehouse stock production for the near future. Production data, gathered from March 2017 until June 2017 shows that on average 25.15 mixed standard products are made per day. The corresponding standard deviation is 3.7 products. These results are calculated using (3.1). During time of this research it was not possible to collect data for the suction irrigation instrument because this instrument was not produced.

However, the production of instruments and suction irrigation devices is done separately. Only one eleventh of total production is used to produce the suction irrigation device, due to the fact that every surgery consumes 5 standard instruments and one in two surgerys a suction irrigation device. Therefore, the total production time can be split in time for standard build and production time for suction irrigation build.

In total 254 days out of one year can be produced which corresponds to 231 days for standard bodies and 23 days for suction irrigation. With this knowledge, the average daily demand per product can be calculated. For example, sales scenario A 2018, 2750 procedures per year corresponds with $\frac{2750\times5}{231}=59.52$ standard products per day. As explained above, currently the average daily demand is 25.15 products with 3.7 standard deviation. These values can be extrapolated to the desired demand using (4.2).

$$\mu \text{ and } \sigma \text{ correspond to } n\cdot\mu \text{ and } \sqrt{n}\cdot\sigma \tag{4.2}$$

The average demand for sales scenarios A of 2018 is 59.52 products and the standard deviation is $n = 59.52/25.15 = 2.36$, $\sqrt{2.36} \times 3.7 = 5.69$. All average demand and standard deviations per sales scenario are given in Table 4.9. Because there is not enough production data available for the suction irrigation, the production data of the standard products are used to give an estimate of demand and standard deviation.

| Scenario A Year | 2017 | 2018 | 2019 |
|---|---|---|---|
| # Procedures | 780 | 2750 | 5500 |
| # Standard Instruments per year | 3900 | 13750 | 27500 |
| # Suction Irrigation per year | 390 | 1375 | 2750 |
| Demand standard instruments per day | 16,88 | 59,52 | 119,05 |
| Standard deviation standard per day | 3,03 | 5,69 | 8,05 |
| Demand SI per day | 16,96 | 59,78 | 119,57 |
| Standard deviation SI per day | 3,04 | 5,70 | 8,07 |

| Scenario B Year | 2017 | 2018 | 2019 |
|---|---|---|---|
| # Procedures | 780 | 2750 | 4125 |
| # Standard Instruments per year | 3900 | 13750 | 20625 |
| # Suction Irrigation per year | 390 | 1375 | 2062,5 |
| Demand standard instruments per day | 16,88 | 59,52 | 89,29 |
| Standard deviation standard per day | 3,03 | 5,69 | 6,97 |
| Demand SI per day | 16,96 | 59,78 | 89,67 |
| Standard deviation SI per day | 3,04 | 5,70 | 6,99 |

Table 4.9: Calculation of the daily demand and standard deviation of all sales scenarios

# Chapter 5

# Elaboration of Inventory Policies

This chapter provides results of the theoretical policies given in Section 3.5. First the theoretical optimum values are calculated for the current demand of the production line. Then the theoretical optimum values of the forecasts are given.

## 5.1 $(Q, s)$ Policy

This section provides the results for the $(Q, s)$ policy. First the theoretical inventory policy is determined. A matlab script with input file is established to calculate the optimal inventory costs with reorder levels and order quantities. Followed by a calculation which determines the average amount of full time employees per day to handle all orders. Then the $(Q, s)$ policy with restricted order quantities is determined. This implies that the order quantities have a maximum limit. At last, the optimal inventory costs with and without order restriction for the forecast of 3 years are calculated with the required FTE's.

### 5.1.1 Theoretical

To calculate the theoretical optimum values of the order quantity, $Q$, and order level, $s$, several parameters have to be set. The daily demand and standard deviation of the current state and the forecast of the production line are given in Section 4.7. Take into account that the given demands are expressed in products per day only if the corresponding product type is produced. Additionally not all inventory parts are used in all products resulting in a decreased demand for these parts.
The following parameters are used to calculate the optimum values of $Q$ and $s$ for a component which is used in all standard instruments, namely buy-in part 401218. Another reason why this example uses this part is because the lead time is not dependent of the batch size and therefore the policy can be calculated theoretically:

- Demand = 25.15 products per day
- Standard deviation = 3.7 products per day
- Lead time = 2.15 hours
- Demand during lead time = 6.72 products
- Standard deviation during lead time = 1.92
- Penalty costs = 139.32 euro per bin
- Order costs = 2.16 euro per order
- Holding costs = 0.000148 euro per day

Using all equations shown in Section 3.5 the optimal $Q$ and $s$ for part 401218 is calculated. The optimal order quantity $(Q^*)$ = 857, the optimal reorder point $(s^*)$ = 18 with the corresponding daily inventory costs= 0.065, replenishment costs= 0.063 and shortage costs= $6.45e^{-5}$. The holding costs are relatively low compared to the order costs which results in a large order quantity.

A matlab script is established to calculate all optimal $Q$ and $s$ values for all components. The complete matlab script for the $(Q, s)$ policy is shown in Appendix A2.1. Firstly matlab uses an input file with all required parameters to calculate the $(Q, s)$ policy. This input file is given in Appendix A2.2. As can be seen, demand and standard deviation per day is requested. The lead times calculated in Table 4.6 are given as input parameter in this file. Because the lead time is order quantity dependent, the lead time is divided in two parts, namely the minimal lead time per order and the size dependent lead time. Also the penalty costs are order quantity dependent, see Table 4.7. The penalty costs can be expressed as the amount of operators effected by a stock out multiplied with the corresponding lead time for that product and the operator costs per hour. Also the calculated order cost per bin, item costs and holding costs shown in Section 3.3 are used as input parameters. All parameters shown in the input file can be adjusted when needed. The demand and standard deviation per day per product is automatically calculated using (4.2) when the daily demand and standard deviation are given. For example, the daily demand of standard products is 25.15 products and standard deviation 3.7. The demand per day for part 401134 can be calculated. Part 401134 is only used in 80% of the products of one day. Using (4.2), $n = 0.8$ results in a daily demand of $0.8 \cdot 25.15 = 20.1$ and a standard deviation of $\sqrt{0.8} \cdot 3.7 = 3.31$. As can be seen in the input file in Appendix A2.2, the demand per day with ratio and standard deviation with ratio of part 401134 corresponds to 20.1 and 3.31 respectively.

The matlab script is divided into several parts. Because some parameters are order quantity dependent, a range for possible order quantities is set. Using this range it is possible to calculate the corresponding order costs per part. Initially the maximum volumes of the bins are not restricted, this results in the optimal order quantities and order levels per part.

```
1 Q = [1:1:2000];
2 for ii=1:length(a);
3    for jj = 1:length(Q);
4        K(ii,jj)= ceil(Q(jj)/Batchgrootte(ii))*Ordercost(ii);
5
6 leadtime(ii,jj)= lt(ii) + (Var_leadtime(ii).*Q(jj)) ;
7 mu(ii,jj)= (leadtime(ii,jj)./hours_day).*a(ii);
8 sigma(ii,jj) = sqrt((var_day_type(ii).*mu(ii,Q(jj)))./a(ii));
9 pi(ii,jj)= leadtime(ii,jj).*penaltyp(ii).*OperatorCost ;
```

As can be seen above, in a snippet of the matlab code of Appendix A2.1, firstly the order costs per part per order quantity range are calculated. Due to the fact that bin sizes are not restricted, the order costs are equal to ordering one batch. The next step is to calculate all corresponding lead times per order quantity range. The lead times depend on a fixed part and a variable part as can be seen in line 6 of the snippet. When the lead time is known it is possible to calculate the demand and standard deviation during lead time, using line 7 and 8 of the snippet. As mentioned before, also the penalty costs are order quantity dependent proportional to the lead time. Line 9 of the snippet calculates the corresponding penalty costs per part per order quantity range.

```
1 dKs(ii,jj)= 1- (h(ii)*Q(jj))/(pi(ii,jj)*a(ii)) ;   % P stock out
2 Ks(ii,jj)= norminv(dKs(ii,jj),0,1)  ; % returns inverse cdf, Z value
3 s(ii,jj) = mu(ii,jj) + (Ks(ii)*sigma(ii,jj))    ;
```

When all parameters are calculated for the order quantity range, the corresponding optimum reorder points can also be calculated. This is done in lines 39 – 44 shown in Appendix A2.1 and shown in the snippet above. If a stock out occurs, the penalty costs are independent of the amount of shortage. As explained in Section 3.5.1, shortage occurs if demand is larger than the reorder point and can be expressed as (3.15) – (3.19). The probability that demand is smaller than the reorder point is calculated in line 1. The value of the probability of not lacking is searched in the normal distribution tables. The corresponding c.d.f. value will return the $Z$ value in line

2. This $Z$ value is used to set the reorder level (line 3), and ensures the corresponding service levels.

When all order points are calculated for the order quantity range the corresponding inventory costs, replenishment costs and shortage costs are calculated using (3.8), (3.9) and (3.10). A summation of these costs results in the total costs per day for the policy for the range of order quantity. Figure 5.1 shows the graph of the total costs per day for the order quantity range for product 1. Finally the script determines the minimal value of the total costs for each product with

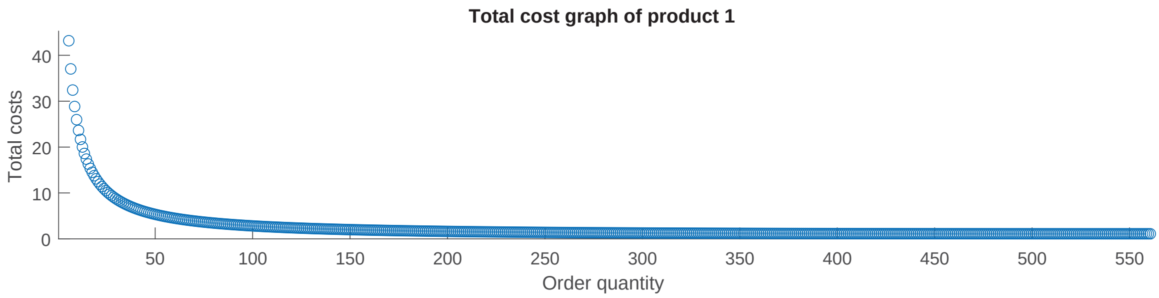


Figure 5.1: Total costs per day for the order quantities for product 1

the corresponding order quantity. Knowing the optimal order quantity the corresponding optimal reorder level, inventory costs, replenishment costs and shortage costs are established. All results of the optimal order costs per component are displayed in Appendix A2.3 in Figure A2.3.

Figure A2.3 in Appendix A2.3 shows the output table created by the matlab script for all components. The first column displays the component numbers. Numbers which start with a star are used for suction irrigation devices. All other components without a star are used in the standard instruments. The mix of standard product types is predefined in the input file (ratio) and included in the average demand. The second column shows the total average cost per day per component. The third and fourth column show the optimal order quantity and optimal reorder level respectively. Column five until seven show the average inventory cost, average replenishment cost and average shortage cost, respectively. The eighth column shows the average amount of orders per day for the corresponding part given the optimal order quantity. This value is calculated using the replenishment costs per day divided by the total order costs per part per day. The total order costs consist of the order costs per batch multiplied with the amount of batches ordered. The last column gives the service levels per component, which corresponds to the proportion of demand that is satisfied from inventory in percentages.

If the optimal costs per component are known the total optimal costs per day for production of the standard instruments (standard) and the total optimal costs per day for production of the suction irrigation device (SI) can be determined. Both values are independent and only apply if the corresponding product type is produced. Figure 5.2 shows the total sum of all optimal costs per component per product type.

| | Standard | SI |
|---|---|---|
| Total Costs per day per product type | 18.1535 | 14.0941 |

Figure 5.2: Output of total costs per product type of matlab script for $(Q, s)$ policy

The optimal costs for the $(Q, s)$ policy of the standard instrument production is equal to 18.15 euros. As mentioned before, the production costs and processing costs of parts are included in the product costs, therefore these $(Q, s)$ policy costs are an addition to the production and processing costs.

As can be seen in Figure A2.3 in Appendix A2.3, the optimal order quantities are very large. Due to the fact that the volume of the orders is not restricted, the optimal order quantities could be very large. In the next section, the amount of full time employees necessary to handle the orders is calculated. Then the input parameters are adapted to calculate the optimal $(Q, s)$ policy if the orders are volume restricted. Followed by the optimal solutions for the next three years of forecast.

### 5.1.2 (Q,s) Policy Costs with FTE

The previous section gave an insight in the optimal cost per component and per product type. Besides the inventory policy costs, the amount of replenishment time per day is also an important number to check whether it is possible for the available amount of operators to handle all orders. In this section the average needed operator time per day is determined.

The replenishment time is also dependent of the type of products produced. The average amount of replenishment per day per part is already calculated in the previous section and shown in the eighth column of Figure A2.3. Each order requires a specific amount of operator time. Fortimedix has two types of operators which can handle the orders, one type handles process M1 of the manufacturing components. And one type of operator handles the buy-in components and processes M2 and M4. Table 5.1 shows the corresponding operator time per operator per component.

| Component | Type A Operator Actions per order | Type A Operator Actions per day | Type B operator Actions per order | Type B Operator operator Actions per day |
|---|---|---|---|---|
| Buy-in Components | none | none | *Counting products required in a bin (2 min)<br>*Stay at process B2 (5 min) and provide bin with tracability information simultaniously<br>*Put components in process M3 and take out when finished. Put components in bin (1 min) | *Pick up empty bins from clean room (10min)<br>*Bring orders to clean room (10 min) |
| Manufactured Components | *Make batch card (1min)<br>*Finish batch (1min)<br>*Transport finished orders to process M2 (5min) | | *Actions to be taken for process M2 (5min)<br>*Transport and finished batches to process M4 (7 min)<br>* Process M4 Time (M4 process time per part × batch size (Q_star)) | |

Table 5.1: The operator time per type of order and operator

As can be seen in Figure A2.3 the average amount of replenishment per part per day is available. Combining the information of Figure A2.3 and Table 5.1 it is possible to calculate the average operator time per day per component. For example, part 401052 is ordered 0.0649 times per day. The operator time for part 401052 is 8 minutes which results in a average required operator time of $0.0649 \times 8 = 0.5192$ minutes per day. If part 401052 is the only product ordered, the total operator time equals $20 + 0.5192 = 20.5192$ minutes per day because the fixed operator time per day is 20 minutes.

To calculate the amount of replenishment, the matlab script is adapted. The operator time per component per operator and the process M4 times per part are included in the model. See line 92 – 124 of Appendix A2.1 for the expansion of the matlab script. This expansion of the matlab script calculates the required operator time per operator type per part. The result of this expansion is given in Figure 5.3. As can be seen, the average operator time per type B operator is 226.83 minutes and the average operator time per type A operator is 1.44 minutes per day. Due to the fact that only a few parts are handled by the type A operator and the batch sizes are unlimited, the type A operator time is very small. The second column shows the amount of Full Time Employees (FTE) required to handle the operator time. One FTE corresponds to 450 minutes a day.

| | Minutes | FTE |
|---|---|---|
| Average operator B time, Standard | 226.8300 | 0.5041 |
| Average operator A time, Standard | 1.4434 | 0.0032 |
| Average operator B time, SI | 123.8737 | 0.2753 |
| Average operator A time, SI | 0.7545 | 0.0017 |

Figure 5.3: The result of operator time with the corresponding demand of 25.15 products per day

The next step is to calculate the inventory policy with order quantity restriction. Due to operational reasons Fortimedix uses maximum batch sizes per order.

### 5.1.3 Inventory policy order quantity restriction

Previous section gave an insight in the optimal inventory cost policy. However, using these optimal order quantities and order levels is not practical. Currently, process M2 restricts the capacity of an order for the manufacturing flow. And process B2 restricts the order quantity of the buy-in component flow. Therefore, the next step is to calculate the optimal $(Q, s)$ policy with volume restriction. As can be seen in the matlab script in Appendix A2.1, two parameters depend on order size. The order cost and the amount of orders per day. Placing an order that has the size of a maximum batch, the costs are the most optimal. Once an order is slightly larger than the maximum batch size, an extra batch has to be ordered with additional order costs. The order costs significantly influence the optimal cost policy. Figure 5.4 shows the total costs for part 401131 with order quantity restriction. The maximum volume of one batch of part 401131 is 30 pieces. As can be seen in the graph, if one part more is ordered than the maximum batch size, the total costs leap. For this example, the optimal order quantity is 30 pieces. The second parameter which is influenced by the batch size is the amount of orders per day. If the batch sizes decrease the amount of orders has to increase because demand has to be satisfied.

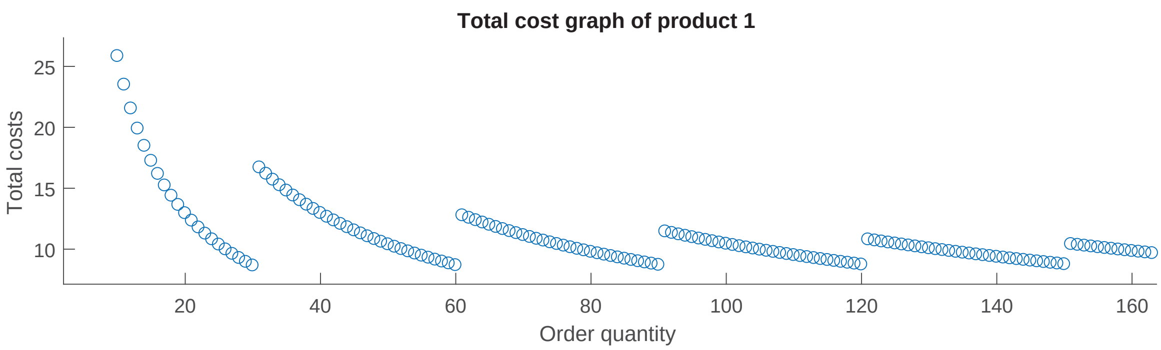


Figure 5.4: Graph of the total costs per order quantity with restricted batch size of part 401131

All maximum bin sizes as shown in Table A1.1 are used as input parameter for the matlab script. The script calculates all optimal order quantities, reorder levels and costs per part. These costs are added per product type and the results of the script are shown in Figure 5.5.

Figure 5.5 shows the total costs per day per product type with a daily demand of 25.15 products with a standard deviation of 3.7. Compared to the answer shown in Figure 5.2 can be seen that the total costs increase significantly. If the order quantity is not restricted, the total costs per day are equal to 18.15 euro for the standard products and 14.09 euro for the suction irrigation.

| | Standard | SI |
|---|---|---|
| Total Costs per day per product type | 62.4056 | 41.4471 |

Figure 5.5: Output of total costs per product type of matlab scrip for $(Q, s)$ policy with restricted order sizes

However, due to operational reasons Fortimedix uses restricted batch sizes. The total costs per day with restricted order sizes are equal to 62.4 euro for the standard products and 41.4 euro for the suction irrigation. This leads to an optimal cost increase of 250% for the standard product and 200% cost increase for the suction irrigation product. If all optimal costs are calculated the required operator time can be determined.

| | Minutes | FTE |
|---|---|---|
| Average operator B time, Standard | 295.7675 | 0.6573 |
| Average operator A time, Standard | 30.8888 | 0.0686 |
| Average operator B time, SI | 168.4688 | 0.3744 |
| Average operator A time, SI | 18.1918 | 0.0404 |

Figure 5.6: The result of operator time with demand 25.15 and restricted order sizes

Figure 5.6 shows the average operator times per type of operator for the restricted order quantities. Compared to the operator times shown in Figure 5.3 can be seen that also the average operator times per operator increase. The increase of the type B operator for the standard product and suction irrigation are 30% and 36% respectively. However, the increase of the type A operator is significant because all orders are divided into small orders which all need operator time. The increase of the type A operator for the standard product and suction irrigation are 2000% and 2300% respectively. As can be seen, restricting the order quantities results in a significant increase of order costs and operator times.

### 5.1.4 Inventory Policy Costs forecast

The next step is to calculate all inventory costs per forecast scenario (displayed in Table 4.8) with the corresponding amount of FTE. Using the adapted matlab script the results of the next three year forecast are shown in Table 5.2. The optimal $(Q, s)$ policy for non restricted order quantities and restricted order quantities are given. The columns with an 'R' behind the year are the results of the restricted order quantities.

As can be seen in the table, the total daily inventory costs increase every year due to the fact that the demand increases. However, the total cost do not increase with the same ratio as the demand. Also can be seen that the difference between the restricted and non-restricted optimal costs significantly increase if the demand increases. For example in 2019 the daily demand is 119 products (Table 4.9), the optimal total $(Q, s)$ policy costs result in 27.1 euro per day. However, if the optimal order quantities are restricted to the maximum bin sizes, the optimal total $(Q, s)$ policy costs increase significantly to 251.5 euros per day. Also the amount of type B operator FTE increases with 35% and the amount of type A operator FTE increases with 4500%. It is recommended to choose the maximum batch size correctly to minimize the costs and FTE. However extreme large batch sizes for the manufactured components are also not practical, large batch sizes result in long lead times. Also the check weather the manufactured components are approved for final assembly takes longer. As stated before, it is assumed that process M1 supplies all required components without rejections to process M2 to calculate the inventory policy. However, in practice it is possible that a batch is rejected at process M1. Then a complete new batch must be processed and all next scheduled orders for process M1 are on hold. If the batch sizes are large,

| Forecast Scenario A | 2017 | 2017R | 2018 | 2018R | 2019 | 2019R |
|---|---|---|---|---|---|---|
| Total daily inventory $(Q, s)$ policy costs Standard instrument | €16.8 | €45.7 | €22.2 | €131.6 | €27.1 | €251.5 |
| Average type B FTE Standard | 0.36 | 0.46 | 1.12 | 1.49 | 2.18 | 2.95 |
| Average type A FTE Standard | 0.003 | 0.046 | 0.005 | 0.16 | 0.007 | 0.32 |
| Total daily inventory $(Q, s)$ policy costs Suction Irrigation | €13.5 | €31.5 | €15.9 | €83.4 | €18.1 | €155.9 |
| Average type B FTE SI | 0.20 | 0.27 | 0.59 | 0.83 | 1.12 | 1.61 |
| Average type A FTE SI | 0.0014 | 0.027 | 0.0026 | 0.096 | 0.0039 | 0.19 |

Table 5.2: Optimal $(Q, s)$ inventory policy costs with corresponding required amount of FTE

the amount of rejected components is also large and all components have to be processed again resulting in a large disruption of process M1. Therefore, it is important to choose the batch sizes correctly. As stated before, process M2 and B2 restrict the maximum order quantities. However, the maximum order quantities are estimated and could change after determination of the exact maximum quantities. Also it is possible that the capacity of the processes change which also results in adapted maximum order quantities. Therefore, the matlab script which calculates the optimal $(Q, s)$ policy costs is compiled and can be used as a calculation tool. This calculation tool returns the optimal costs per component, the total optimal costs per product type and the corresponding required FTE's. Also the graph for the optimal costs per part can be displayed. All parameters of the input file, shown in Appendix A2.2 can be adapted to obtain the desired $(Q, s)$ policy results.

## 5.2 $(R, S)$ Policy

This section provides the results for the $(R, S)$ policy. The layout of this section is equal to the layout of the previous section. First the theoretical inventory policy is determined. A matlab script with input file is established to calculate the optimal $(R, S)$ policy and the required FTE's are determined. For this policy also the batch restriction is applied and the forecast of inventory costs of 3 years with required FTE's is given.

### 5.2.1 Theoretical

To calculate the theoretical optimum reorder level, $S$, several parameters have to be set. The daily demand and standard deviation of the current state and the forecast of the production line are given in Section 4.7. The following parameters are used to calculate $S$ for part 401218:

- Demand = 25.15 products per day
- Standard deviation = 3.7 products per day
- Lead time = 2.15 hours
- Review time = 1 day
- Demand during lead time + R = 31.9 products
- Standard deviation during lead time + R = 4.17
- Penalty costs = 139.32 euro per bin
- Order costs = 2.16 euro per order
- Holding costs = 0.000148 euro per day.

Using all equations shown in Section 3.5 the optimal reorder level ($S$) for part 401218 is found. The optimal order level is 52 with the corresponding inventory costs of 0.0047, replenishment costs of 2.16 and stock out costs of $1.435e^{-6}$ euro per day. Due to the relatively low holding costs, a small review period is not practical. Therefore, the next step is to find the optimal $(R, S)$ policy with the corresponding review period per part.

Similar to the $(Q, s)$ policy, it is possible to make a matlab script to calculate the optimal $(R, S)$ policy. This script is similar to the script of the $(Q, s)$ policy. The complete matlab script for the $(R, S)$ policy is shown in Appendix A3.1. The difference between the $(Q, s)$ policy and $(R, S)$ policy is that the $(R, S)$ policy does not use a fixed order quantity. Because the order lead time and the penalty costs depend on the order quantity Q, an initial order quantity has to be set to calculate the $(R, S)$ policy. Therefore, the maximum volumes per bin are used as initial order quantities. This could initially result in larger lead times and larger penalty costs which will result in a higher average stock level.

Firstly matlab uses an input file with all required parameters to calculate the $(R, S)$ policy. This input file is given in Appendix A3.2. As can be seen, demand and standard deviation per day is requested but a fixed review period is not required to enter. Also this matlab script is divided in several parts. Some parameters are review period dependent, therefore a range for possible review periods is given. The matlab script checks whether a fixed review period is given in the input file. If the review period is given, the script calculates the $(R, S)$ policy for that given review period. If this value is blank, the script calculates all policy costs for a range of review times and calculates the optimal $(R, S)$ policy with the corresponding review period for all components.

Initially the maximum volumes of the bins are assumed not restricted, 2000 parts per component, this will result in the optimal reorder levels per component.

```
for jj = 1:length(R);
for ii=1:length(a);

leadtime(jj,ii)= lt(ii) + (Var_leadtime(ii).*Batchgrootte(ii)) ;
mu(jj,ii)= (R(jj)+(leadtime(ii)./hours_day)).*a(ii);
sigma(jj,ii)= stdv_day_type(ii)*sqrt(R(jj)+(leadtime(jj,ii)./hours_day
    ));
pi(jj,ii)= leadtime(ii).*penaltyp(ii).*OperatorCost ;
```

As can be seen in Lines 34 – 40 of Appendix A3.1 and in the snippet above, the average demand and standard deviation during lead time and review period are calculated. Followed by the penalty costs per lacking component order. With these parameters it is possible to calculate the probability of stock-out. If a stock-out occurs, the penalty costs are independent of the amount of shortage. As explained in Section 3.5.2, shortage occurs if demand is larger than the reorder level ($S$) and can be expressed as (3.28).

```
1 dKs(jj,ii)= 1- ((h(ii).*R(jj))/(pi(jj,ii))); % P stock out
2 Ks(jj,ii)= norminv(dKs(jj,ii),0,1); % returns inverse cdf, Z value
3
4 % Calculate reorder level and safety stock
5 S(jj,ii) = (mu(jj,ii)+(sigma(jj,ii)*Ks(jj,ii)));
6 Cs(jj,ii) = (1 - dKs(jj,ii)); %pi*[1-F(S)]
7 SS(jj,ii) = S(jj,ii)-mu(jj,ii);
```

The probability that demand is smaller than the reorder levels is calculated in line 43 of Appendix A3.1 and line 1 in the snippet above. The values of the probability of not lacking is searched in the normal distribution tables. The corresponding c.d.f. values will return the $Z$ values in line 2. These $Z$ value are used to set the reorder levels and ensures the corresponding service levels.

The probability of shortage is 1 minus the probability of service and calculated in line 6. The safety stocks per component are calculated in line 7 using the reorder levels minus the average demands.

The next step is to calculate the order costs per component. Due to the fact that bin sizes are not restricted initially, the order costs are equal to ordering one batch. If the bin sizes are restricted, the order costs will increase significantly. The order quantities are variable in a $(R, S)$ policy therefore it is not possible to calculate the exact order costs. The order costs depend on the average demand and the bin sizes. The average demand during a review period can be expressed as the reorder level minus the safety stock. Dividing the average demand during a review period by the maximum bin size, the amount of bins ordered is obtained. This amount of bins ordered multiplied with the order costs will result in the total order costs during the review period.

When all order levels are calculated for the range of review points, the corresponding inventory costs, replenishment costs and shortage costs are calculated using (3.23) (3.24) and (3.25). A summation of these costs results in the total costs per day for the $(R, S)$ policy for the range of review period. Finally the minimal value of the total costs in the review period range is determined and the script returns the minimal optimal costs per product with the corresponding optimal reorder period. Knowing the optimal reorder period the corresponding optimal reorder level, inventory costs, replenishment costs and shortage costs are established. All results using infinite bin sizes and a range of order periods are displayed in a table and shown in Figure A3.4 in Appendix A3.3.

This table shows the output table created by the matlab script for all components. The first column displays the component numbers. Numbers which start with a star are used for suction irrigation devices and without a star are used in the standard instruments. The second column shows the optimal costs per component per day. The third and fourth column show the reorder levels and safety levels per component. Column five until seven show the daily average inventory cost, average replenishment cost and average shortage cost, respectively. The eighth column shows the average amount of orders per day for the corresponding component given the optimal reorder level and review period. This value is calculated using the amount of replenishment per review period divided by the optimal review period. The ninth column gives the service levels per component, which corresponds to the proportion of demand that is satisfied from inventory in percentages. The tenth column shows the optimal review period in days and the last column shows the amount of bins ordered per review period.

When all optimal costs per component are known, the total cost per day per instrument type can be determined. Figure 5.7 shows the sum of all optimal costs per component for each instrument type. These costs correspond to the optimal costs for the $(R, S)$ policy. Both values are independent and only apply if the corresponding instrument type is produced. The optimal costs for the $(R, S)$ policy of the standard instrument production is equal to 14.13 euros per day.

| | Standard | SI |
|---|---|---|
| Total Costs per day per product type | 19.0988 | 14.4205 |

Figure 5.7: Output of total costs per product type of matlab script for $(R, S)$ policy

Due to the infinite bin sizes, all orders consist of one batch. However when bin sizes are limited the order costs will increase significantly. Comparing the total costs of the $(Q, s)$ policy and the $(R, S)$ policy with infinite bin sizes will result in small differences. The optimum $(Q, s)$ policy costs for the standard product is 18.15 euro and the optimum $(R, S)$ policy costs is 19.09 euro. The next step is to calculate the required amount of FTE to handle the orders. Then the bin sizes are restricted to calculate the optimal $(R, S)$ policy if the orders are volume restricted. Followed by the optimal solutions for the next three year of forecast.

### 5.2.2 (R,S) Policy Costs with FTE

In previous section the optimal inventory costs for the $(R, S)$ policy with unlimited batch sizes are determined. The next step is to calculate the required amount of FTE to handle the orders. As can be seen in Figure A3.4 the average amount of replenishment per part per day is available. Combining the information of Figure A3.4 and Table 5.1 it is possible to calculate the average operator time per day per component. The matlab script in Appendix A3.1 is extended with the same FTE calculation as used for the $(Q, s)$ policy. The only difference with the $(R, S)$ policy is that the exact number of order quantity is not know. The order quantity is needed to calculate the M4 process time per operator. Therefore, an average order quantity per day is calculated using the average demand during lead time and review period divided by the review period in days.

The operator time per component per type of operator and the process M4 times per part are included in the model. This expansion of the matlab script calculates the required operator time per operator type per component. The result of the calculations are given in Figure 5.8. As can be seen, the average operator time per type B operator is 288.45 minutes and the average operator time per type A operator is 1.48 minutes per day. Due to the fact that only a few parts are handled by the type A operator and the batch sizes are unlimited, the type A operator time is very small. The second column shows the amount of FTE required to handle the operator time. One FTE corresponds to 450 minutes a day.

| | Minutes | FTE |
|---|---|---|
| Average operator B time, Standard | 288.4505 | 0.6410 |
| Average operator A time, Standard | 1.4899 | 0.0033 |
| Average operator B time, SI | 156.4733 | 0.3477 |
| Average operator A time, SI | 0.7715 | 0.0017 |

Figure 5.8: The result of operator time with the corresponding demand of 25.15 products per day

The amount of operator time needed to process the optimal $(Q, s)$ policy is smaller than the amount of operator time needed to process the optimal $(R, S)$ policy. The average operator time per type B operator for the $(Q, s)$ policy is 226.83 minutes and the average operator time per type A operator is 1.44 minutes per day. The demand in-between replenishment of the $(Q, s)$ policy only depends on the lead time, and the demand in-between replenishment of the $(R, S)$ policy depends on the lead time and review period. Therefore, the demand of the $(R, S)$ policy has a larger deviation of the uncertain demand which results in higher safety levels, which results in slightly larger inventory costs and replenishment costs.

The next step is to calculate the inventory policy with bin size restriction. Due to operational reason Fortimedix uses maximum batch sizes per order.

### 5.2.3 $(R, S)$ policy order quantity restriction

Previous sections gave an insight in the optimal $(R, S)$ inventory cost policy and the required FTE's. As can be seen in Figure A3.4 in Appendix A3.1, the reorder levels are large. Ordering large quantities of components is not practical due to large lead times and the risk of rejected inventory. Currently, the process M2 restricts the capacity of an order. The next step is to calculate the optimal $(R, S)$ policy with volume restriction.

As can be seen in the matlab script in Appendix A3.1, several parameters depend on the bin sizes. The lead time, order costs and the amount of orders per day are bin size dependent. Placing an order that has the maximum bin size results in the optimal order costs. Once an order is larger than the maximum batch size, an extra batch has to be ordered with additional order costs. Another parameter which is influenced by the batch size is the amount of orders per day. If the batch sizes decrease, the amount of orders has to increase because demand has to be satisfied. All maximum bin sizes as shown in Table A1.1 are used as input parameter for the matlab script. The script calculates all optimal reorder levels, optimal policy costs and required FTE's. These optimal costs are added per product type and the results of the script are shown in Figure 5.9 and Figure 5.10.

| | Standard | SI |
|---|---|---|
| Total Costs per day per product type | 66.1784 | 43.5575 |

Figure 5.9: Output of total costs per product type of matlab scrip for $(R, S)$ policy with restricted order sizes

Figure 5.9 shows the total costs per day per product type with a daily demand of 25.15 products with a standard deviation of 3.7. Compared to the answer shown in Figure 5.7 can be seen that the total costs increase significantly. If the order quantity is not restricted, the total costs per day are equal to 19.09 euro for the standard products and 14.42 euro for the suction irrigation. The total costs per day with restricted order sizes are equal to 66.2 euro for the standard products and 43.5 euro for the suction irrigation. This leads to an optimal cost increase of 250% for the standard product and 200% cost increase for the suction irrigation product.

| | Minutes | FTE |
|---|---|---|
| Average operator B time, Standard | 303.1417 | 0.6736 |
| Average operator A time, Standard | 31.7806 | 0.0706 |
| Average operator B time, SI | 172.6032 | 0.3836 |
| Average operator A time, SI | 18.6796 | 0.0415 |

Figure 5.10: The result of operator time with demand 25.15 and restricted order sizes

Figure 5.10 shows the average operator times per type of operator for the restricted order quantities. Compared to the operator times shown in Figure 5.8 can be seen that also the average operator times per operator increase. The increase of the type B operator for the standard product and suction irrigation are 5% and 10% respectively. However, the increase of the type A operator is significant because all orders are divided into small orders which all need operator time. The increase of the type A operator for the standard product and suction irrigation are 2000% and 2400% respectively. As can be seen, restricting the order quantities for operational reasons results in a significant increase of order costs and operator times.

### 5.2.4 $(R, S)$ Inventory Policy Costs forecast

In the previous sections the optimal inventory costs for the $(R, S)$ policy are calculated for restricted and non restricted batch sizes. Also the corresponding amount of FTE are calculated. The next step is to calculate all inventory costs per forecast scenario (displayed in Table 4.8) with the corresponding amount of FTE. Using the extended matlab script the results of the next three year forecast are determined and shown in Table 5.3. The optimal $(R, S)$ policy for non restricted order quantities and restricted order quantities are given. The columns with an 'R' behind the year are the results of the restricted order quantities.

As can be seen in the table, the total daily inventory costs increase every year but demand also increases every year. However, the total cost do not increase with the same ratio as the

| Forecast Scenario A | 2017 | 2017R | 2018 | 2018R | 2019 | 2019R |
|---|---|---|---|---|---|---|
| Total daily inventory $(R, S)$ policy costs Standard instrument | €17.7 | €48.3 | €23.3 | €140.1 | €29.8 | €267.8 |
| Average type B FTE Standard | 0.43 | 0.47 | 1.6 | 1.54 | 4.14 | 3.03 |
| Average type A FTE Standard | 0.003 | 0.05 | 0.005 | 0.17 | 0.01 | 0.33 |
| Total daily inventory $(R, S)$ policy costs Suction Irrigation | €13.8 | €32.9 | €16.3 | €88.4 | €19.6 | €165.4 |
| Average type B FTE SI | 0.24 | 0.27 | 0.85 | 0.85 | 2.1 | 1.7 |
| Average type A FTE SI | 0.0014 | 0.03 | 0.003 | 0.01 | 0.005 | 0.2 |

Table 5.3: Optimal (R,S) inventory policy costs with corresponding required amount of FTE

demand. Also can be seen that the difference between the restricted and non-restricted optimal costs significantly increase if the demand increases. For example in 2019 the daily demand is 119 products (Table 4.9), the optimal total $(R, S)$ policy costs result in 29.8 euro per day. However, if the optimal order quantities are restricted to the maximum bin sizes, the optimal total $(R, S)$ policy costs increase significantly to 267.8 euros per day. The amount of required type A operators increases with 3200% but the amount of required type B operators decreases with 27%. The total cost for the $(R, S)$ policy are minimized, if the batch size is not restricted and holding costs are low, the reorder levels are high. This calculation returns the optimal inventory costs with an optimal review period. Therefore, it is not possible to compare the yearly scenarios of the $(R, S)$ policy with or without batch size restriction. However, it is possible to compare the $(R, S)$ policy and the $(Q, s)$ policy with or without batch size restriction.

Fortimedix uses order quantity restrictions. However these maximum amount of products in a bin can be changed in the future. Therefore, the matlab script which calculates the optimal $(R, S)$ policy costs is compiled and can be used as a calculation tool. This calculation tool returns the optimal costs per component, the total optimal costs per product type and the corresponding required FTE's. Also the graph for the optimal costs per part per review period can be displayed. All parameters of the input file, shown in Appendix A3.2 can be adapted to obtain the desired $(R, S)$ policy results.

This chapter showed the elaboration of the $(Q, s)$ and the $(R, S)$ policy. First a theoretical example was given for both scenarios, followed by the elaboration of a calculation tool which calculates the optimal inventory policies. The amount of required FTE to handle the orders are calculated and the forecast scenarios are given. The next step is to validate the matlab script using Simevents.

# Chapter 6

# Simulation Model

The previous chapter showed the elaboration of the $(Q, s)$ and the $(R, S)$ policy. A calculation tool which calculates the optimal inventory policies was created. This calculation tool returns the optimal reorder levels and order quantities for the $(Q, s)$ policy and the optimal order levels and review period of the $(R, S)$ policy. All associated costs of the inventory policies are determined with the required amount of FTE to handle the orders. In this chapter, a Simulink Simevents model is created to perform simulations in order to check whether the calculation tool created in previous chapter can be used to determine the optimal inventory policies at Fortimedix. First a global model of the assembly line is created followed by the supply of components.

## 6.1 Simulation Model

The next step of this research will be creating a simulation model of the assembly line in Matlab Simevents. This model will indicate whether the calculation tool created in previous chapter is valid. Followed by an extension of the assembly line with supply of components.

### 6.1.1 Model Setup

The first step of the simulation model is to translate the assembly line shown in Figure 6.2 to a model which can be configured in Simevents. The translation of the assembly line is given in Figure 6.1. A generator (G) generates jobs which are used to manufacture instruments. These jobs are instantaneously used at the first process of the assembly line (A1). As can be seen, a buffer with components is located at process A1. The components derive from the component manufacturing flow. In between every process a buffer is located to handle the differences in process times. Also process A2 needs components derived from the buy-in component flow.

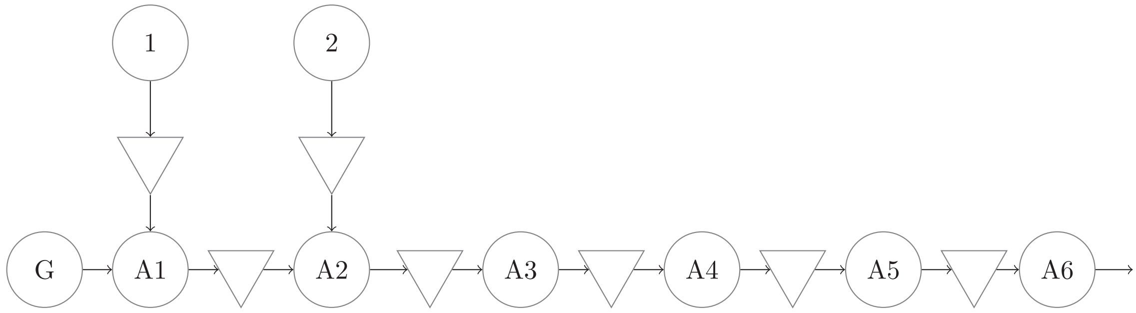


Figure 6.1: Complete process flow

Firstly the simulation model is configured without component supplies. Jobs enter the system and are processed at the workstations. The two main processes to model are the process A1 and A2 because these are the only two processes which need additional components to finish the process. These two processes are modeled and shown in Figure 6.2. As can be seen, a generator generates jobs which are processed at the consecutive processes. Processes are represented by a circle with its corresponding process time and capacity. Detailed information about the modeled process times is given in the next section. Process A1 consists of two workstations, the generated jobs are queued and processed at the first available station. Process A2 consists of 3 operations of which the first and last operation are performed by an operator. Therefore, this process is split in three single operations with intermediate buffers. These intermediate buffers have a maximum capacity due to lack of storage space. The last symbol represents the entity terminator which accepts and destroys entities which are finished processing. Running this simulation returns the maximum amount of products made per day, shown in the display.

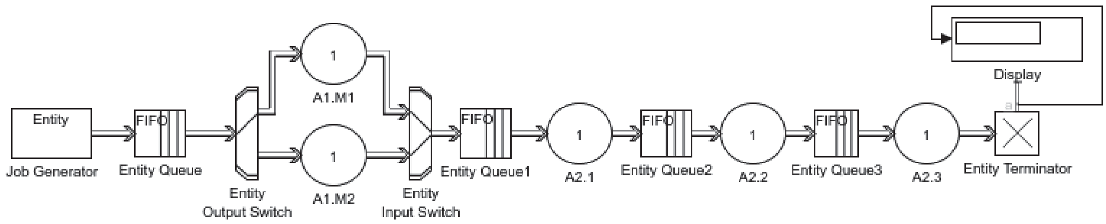


Figure 6.2: Model of the first two processes of the assembly line

## 6.2 Process time modeling with gamma distribution

In previous section a layout of the simulation model is given. The next step is to gather the parameters needed for the processes. The process times can be considered to have a shifted gamma distribution due to the fact that all processes have an absolute minimal process time. The mean and standard deviation of all processes are known and the absolute minimal process time per process are determined. All process times can be expressed as a shifted gamma distribution.

The shifted gamma distribution consists of three parameters, $a$ the shape parameter, $b$ the scale parameter and $\theta$ threshold. Using these symbols, the shifted gamma density function can be expressed as[5]:

$$f(x) = \frac{1}{b\Gamma(x)} \left(\frac{x-\theta}{b}\right)^{a-1} \exp\left(-\frac{x-\theta}{b}\right) \tag{6.1}$$

The mean and variance of the distribution can be expressed as:

$$\mu = \theta + ab \qquad \sigma^2 = ab^2 \tag{6.2}$$

Because the mean, variance and threshold of the process times are known it is possible to calculate the shape and scale parameter using:

$$a = \left(\frac{\mu-\theta}{\sigma}\right)^2 \qquad b = \frac{\sigma^2}{\mu-\theta} \tag{6.3}$$

All parameters of the gamma distributions are summed in Table 6.1.

To model the process times in simevents a small matlab script is written. This script uses a seed causing the simulation to be reproducible. A vector of random gamma distributed values with the

| Process | Mean $\mu$ [min] | Variance $\sigma^2$ [min] | Treshold $\theta$ [min] | Scale parameter $b$ | Shape parameter $a$ |
|---|---|---|---|---|---|
| A1 | 11,75 | 1,93 | 7 | 0,406316 | 11,69041 |
| A2 | 9,92 | 2,7225 | 5 | 0,553354 | 8,89124 |
| A2.1 | 7,35 | 2,95 | 4 | 0,880597 | 3,804237 |
| A2.2 | 2,57 | 0,89 | 0,5 | 0,429952 | 4,814494 |
| A3 | 8,0015 | 1,601 | 5 | 0,5334 | 5,627109 |
| A4 | 1,017 | 0,008595 | 0,6 | 0,020612 | 20,23141 |
| A6 | 1,6025 | 0,293 | 1 | 0,486307 | 1,238929 |

Table 6.1: Process time parameters of the gamma distributed process times

corresponding process time parameters is established. Every time a new job enters the process, the next process time of the vector is chosen. The script used to model the process times for process A1 is given in Figure 6.3 and the histogram of the created process times is given in Figure 6.4. The mean and variance of the data shown in the histogram is equal to 11.756 and 1.93 respectively. This corresponds to the mean and variance of process A1.

```
persistent times x rngInit
if isempty(rngInit)
    seed = 12345;
    rng(seed);
    rngInit = true;
end

if isempty(times)
   times = [gamrnd
       (11.69041,0.406316,100000,1)
       +7]/60;
   x = 1;
end
dt = times(x);
x = x+1;
```

Figure 6.3: Script to generate the gamma distribution for process A1

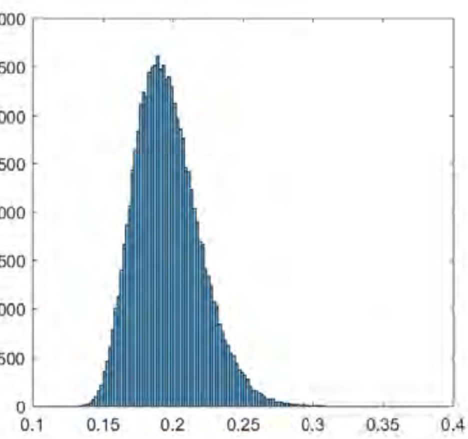


Figure 6.4: Histogram of the gamma distributed service times

## 6.3 Operators

In previous sections the layout of the simulation model is established with the corresponding process times. As can be seen in Table 6.1 the mean process times of process A1 and A2 do not differ much. Initially process A1 was the bottleneck due to the larger process time. However, due to an extra workstation at process A1 the new bottleneck is process A2. To balance the production line, the second workstation of process A1 is a flexible workstation which will only manufacture products when needed. The operator located at workstation 2 of process A1 is also available for other processes. The next step of the simulation is to model the operators in the simulation model. Operators can be modeled using resource pools. The resource pool block defines resources which can be used by entities during the simulation. An example of a resource pool is shown in Figure 6.5. This resource pool represents the amount of operators which can be used for process A2. As can be seen, it is possible to extract the amount of resources in use, u, and the average utilization of the resources, util. To use the resource at a process, an acquirer block is used. After processing the resource is released using a releaser block. The implementation of the acquirer block and releaser block are shown in Figure 6.6.

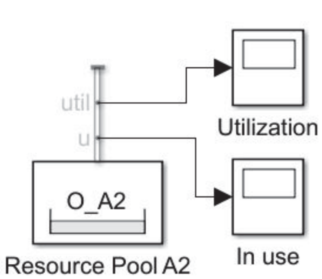


Figure 6.5: The amount of operators for process A2 modeled as resource pool

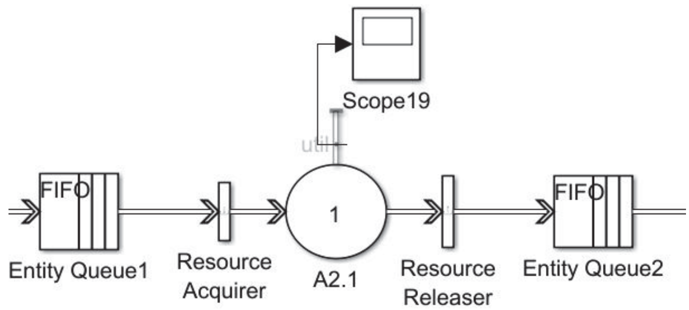


Figure 6.6: The implementation of the recourse acquire and release block

Because the amount of operators used in the assembly line is not infinite, all operators are modeled. Currently Fortimedix uses three operators to handle the daily demand of approximately 25 instruments per day. The implementation of operators in the total assembly line is shown in Appendix A4.1. This model consists of four resource pools with operators which are used at the workstations. As can be seen, process A2.2 and process A5 do not require an operator. Process A5 can be seen as a large buffer with a large delay time, therefore, the capacity of this process is infinite.

This section gave an insight into modeling with resource pools. The next step is to model more practical details into the simulation model.

## 6.4 Simulation features

The previous sections gave an insight into the global simulation model of the assembly line. Resource pools were inserted in the model to create a realistic use of operators. The next step is to simulate more features into the simulation which result in a more realistic simulation. The following features are built in the simulation model:

- Regulate the amount of workstations of process A1
- Insert the amount of desired products per day

- Insert a clock which will regulate a new day in the simulation which will makes it possible to simulate more days successively
- Different product types

The first three features can be simulated using entity gates. A gate can regulate the control of entities via its changing status open or close. Entities are able to pass a gate with open status as long as the subsequent block accepts the entity. A gate with closed status blocks the entity. The status of a gate can be regulated using routing messages. Routing messages can be summoned based on user input or when a programmed event occurs.

Regulating the amount of workstations of process A1 can be done adding an entity gate in front of the second workstation of process A1. If this gate is closed only one workstation is used. The production planning of Fortimedix indicates how many workstations are used. Before the simulation starts, the user indicates whether the gate is open or closed. Appendix A4.2 shows part of the simulation model which regulates the amount of workstations in process A1. As can be seen, Block A1M2(cmd) regulates the status of the entity gate in front of the second workstation of process A1. The routing message A1M2 is programmed inside the block parameters of the TimeCount block. The script inside of the block parameters is given in Appendix A4.2 Figure A4.3. As can be seen, to enable one workstation at process A1 the gate command is zero.

Figure A4.3 also shows that another feature is built in this block. This feature regulates the start of a new day. The amount of entities, in this case minutes, which enters the TimeCount block is added. When the amount of entities is equal to 450 minutes the routing message 'JobDay' is enabled. As can be seen in Figure A4.2, the 'JobDay' routing messages regulates the jobs which enter process A1. This feature ensures that every time a new day starts, new jobs are processed.

The next feature is to regulate the start up amount of jobs per day. This is also regulated using the 'JobDay' routing messages. The queue in front of the entity gate regulated by 'JobDay' counts the departed amount of jobs. The script used to count the departed jobs is given in Figure 6.7.

```
persistent jobsDay
if isempty(jobsDay)
    jobsDay = 0;
end

jobsDay = jobsDay + 1;

% Give desired amount of good products
% per day mod(jobsDay,products)
if mod(jobsDay,25) == 0
    JobDay(0);
end
```

Figure 6.7: The script programmed in the queue

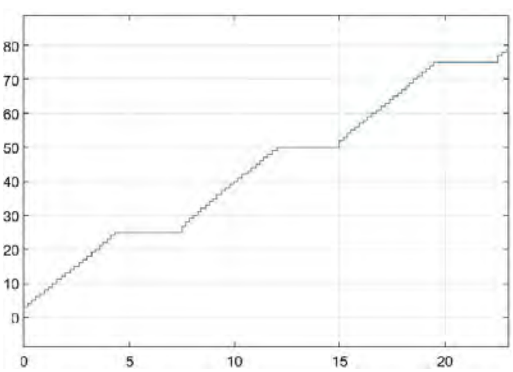


Figure 6.8: The graph of the amount of jobs leaving the queue over time

As can be seen in Figure 6.7, a persistent variable jobsDay is created. If an entity exits the queue, one is added to the variable jobsDay. If the amount of jobs is equal to 25 the routing message JobDay sends a closing signal to the entity gate. As mentioned, also the TimeCount block regulates the entity gate. A combination of these two routing messages will result in the proper amount of start-ups per day. Entities leaving the block Queue are visualized in Scope1 of Appendix A4.2 and shown in Figure 6.8. As can be seen in Figure 6.8, the desired amount of jobs leave the queue. When this amount is reached, no entities leave the queue until a new day starts.

The next step is to model the different types of jobs. Namely the jobs for Scissor, Grasper, Maryland, Clip applier, Hook-knife and the Suction irrigation instrument. This is done using an entity generator for all types of jobs. When a job is created it is possible to give information to a job using entity attributes. Two types of attributes are added to the jobs making it possible to route the jobs through the simulation when necessary. One attribute programmed is the Jobtype and the second attribute modeled checks whether the job is a standard product or not.

## 6.5 Component supply

The next step is to insert the components needed to create an instrument. This is done using the composite entity creator block. This block merges the input entities into a new composed entity. For example, process A1 needs a job entity and 5 different component entities to compose one instrument body entity.

For programming reasons the assembly line is tripled in the simulation model causing each assembly line to manufacture a type of instrument body. The components included in the model are modeled as one entity creator per component because physically there is only one assembly line which uses components from one storage place. The copied assembly lines also use the same resource pools as the original assembly line to make sure that only one of the equally processes is used at the same time. The complete simulation model is shown in Appendix A4.3. Merging Figure A4.4 and Figure A4.5 results in the doubled assembly line and Figure A4.6 shows the suction irrigation assembly line. As can be seen, the upper two assembly lines are equal because the standard body and the hook-knife body experience the same processes, however, with a slight difference in process times. The suction irrigation assembly line is different because the suction irrigation instrument has another routing.

As can be seen, subsystems are created for the supply of components. Initially the component generators are modeled as pre-loaded generators which results in large stock components at simulation start. The script used to model a pre-loaded queue is given in Appendix A4.4. Each process which needs components to manufacture is combined with a subsystem. Several subsystems are directed using routing messages. The status of the routing message is generated in the attributes of the jobs. For example, Figure A4.9 in Appendix A4.5 shows subsystem 4 which is used to supply components to process A2.3. In this subsystem, entities of all required components are created. This subsystem has two outputs, namely the output for the set of components for a standard body or a hook-knife body. As can be seen, an entity input switch uses the input 'In1' to choose the corresponding component using the attribute of the entity. Leading to an assembly of the right set of components.

Modeling these subsystems and component supplies, processes are stopped when a component lacks. The next step is to model the component supply using an inventory policy. However, the simulation model without supply of products can be used to determine lead times of instruments, utilizations of operators and utilizations of processes.

### 6.5.1 Inventory policy for supply of components

The first inventory policy to model is the $(Q, s)$ policy. This policy uses a reorder level and a fixed order quantity. The reorder levels and fixed order quantities per component are derived from the $(Q, s)$ policy calculation tool. Replenishment of a component can be modeled using the amount of entities in a queue. If the amount of entities in a queue is equal to the reorder level of the $(Q, s)$ policy a signal is send to a gate to release a replenishment order. The quantity of the order can be adapted in the order batch creator.

As can be seen in Table 6.1, the average process time of process A1 is 11.75 minutes. This means that approximately 38.29 products ($\frac{450}{11.75}$) per day can be made at process A1 if there are no interruptions. Production data showed (Section 4.7) an average production of 25.15 products corresponds to a standard deviation of 3.7 products. Using (4.2) the standard deviation for a demand of 38.29 products is extrapolated and equal to 6.67 products. Now the daily demand and standard deviation are known for a non-interrupted process. These parameter are inserted into the $(Q, s)$ policy calculation tool. The calculation tool determines the optimal reorder level of 171 products and the order quantity of 673 products. Also the average amount of inventory can be calculated, dividing the inventory cost by holding cost, which results in 394.88 products. And the amount of replenishment, 0.0569 times per day is determined. Appendix A4.6 Figure A4.10 shows the partial model of process A1 with the $(Q, s)$ policy implementation. As can be seen, only process A1 is modeled. If the amount of components in the queue is smaller or equal to 171 a message is send to the entity gate and one bin of 637 components is released. Display 1 shows the average queueing length of 388.2 products and the other display shows the amount of bins ordered. The results of the average queueing length of the simulation are close to the results of the calculation tool, 388.2 and 394.88 respectively. Also the amount of orders is equal. The total simulation time is 10000 hours which is equal to a theoretical replenishment of $\frac{10000 \cdot 0.0569}{7.5}$=75.87 times, and the simulation shows a replenishment of 76 times. This assumes that the calculation tool created to optimize the $(Q, s)$ policy is valid.

The next step is to insert the features created in the partial simulation into the complete simulation model. However, inserting this inventory policy for the supply of components results in a run time error. Therefore, it is not succeeded to validate neither the $(Q, s)$ policy nor $(R, S)$ policy calculation tool on the complete simulation model.

# Chapter 7

# Conclusion and recommendations

This chapter gives the main conclusions of this research. Recommendations for further research on the topic are given.

## 7.1 Conclusion

An insight into the theory of the $(Q, s)$ policy and the $(R, S)$ policy is gathered. Detailed information on processes are investigated and collected to implement the $(Q, s)$ policy and the $(R, S)$ policy.

This research provides the results for the $(Q, s)$ policy and the $(R, S)$ policy. For both policies a calculation tool in matlab is established which calculates the optimal inventory policies.

The calculation tool for the $(Q, s)$ policy calculates the optimal inventory costs, optimal reorder levels and order quantities. The calculation tool for the $(R, S)$ policy calculates the optimal inventory costs, optimal order levels and optimal review period. Both tools also calculate the average amount of full time employees per day which handle all orders. The calculation tools require an input file with parameters to establish the answers. The calculation tools are expanded with bin size restrictions which implies that the order quantities have a maximum limit. All parameters including the maximum bin sizes can be adapted in the input file and the optimal inventory policy is determined. Both calculation tools make it possible for Fortimedix to determine the optimum inventory policy for the desired parameters at every moment. An output graph per component shows the total cost per component per order quantity or review period and can be used to derive the inventory costs for other quantities than optimal values.

A simulation model of the complete assembly line is build in Simevents. The simulation model without supply of products can be used to determine lead times of instruments, utilizations of operators and utilizations of processes. This simulation model is created to perform simulations in order to check whether the calculation tool created in Matlab is valid. However, inserting an inventory policy for the supply of components results in a run time error.

A partial simulation model is established to simulate the $(Q, s)$ policy for one product. The optimal reorder level and order quantity, derived from the calculation tool are used to check whether the calculation tool matches the partial simulation model. The expected average inventory and amount of replenishment of the simulation model is equal to the calculation tool which validates this calculation tool. The validation of the $(R, S)$ policy calculation tool is not succeeded.

The calculations show that bin size restriction has a major influence on the total inventory costs. Not only the costs increase but also the amount of FTE to handle the orders increases significantly. Cost reduction can be realized using the inventory policy and the calculation tool.

## 7.2 Recommendations

The run time error of the complete assembly line simulation results in an incomplete validation of the inventory policies. Therefore, it is recommended to solve the run time error. The first possible step into solving the run time error would be to extend the created partial validation model with multiple components (step by step). Then it is possible to observe when and why an error occurs. If the run time error is solved, it is possible to investigate and compare both inventory policies in the simulation. A clear view of the component supply can be drafted. Resulting in an optimal layout of the component supply, including FTE and machines.

Bin size restriction results in significant increase of total costs. Therefore, it is recommended to implement larger bin sizes. However, excessive bin sizes increase the failure costs in case of rejected supplies. When bin sizes are very large, lead times increase and processes are longer interrupted if a process fails. Therefore, a risk factor for excessive inventory needs to be incorporated in the calculation tool in further research.

This research concludes that implementing inventory policies reduces inventory costs. Both investigated inventory policies can be used at Fortimedix. It is recommended to implement the $(Q, s)$ policy because the total costs are lower and fixed order quantities are preferred.

# Bibliography


[1] REM Associates *Methodology of Calculating Inventory Carrying Costs, pdf* Princeton, New Jersey, last accessed on 07-04-2017 www.remassoc.com/portals/0/remprecc.pdf

[2] S. Nahmias and T. L. Olsen *Production and Operations Analysis, 7th ed* Long Grove, IL: Waveland Press, 2015, p. 206.

[3] P. Jensen and J. Bard *Operations Research Models and Methods S5* url: www.me.utexas.edu/~jensen/ORMM/.../inventory/sq_policy.pdf

[4] P. Jensen and J. Bard *Operations Research Models and Methods S7* url: www.me.utexas.edu/~jensen/ORMM/.../inventory/rs_policy.pdf

[5] NCSS, LLC *Gamma Distribution Fitting* NCSS Statistical Software, Chapter 552 url: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Gamma_Distribution_Fitting.pdf

# Appendix A1

## Costs

This appendix provides additional information for Section 4.4. This appendix shows the item costs per component. The holding cost are calculated based on a yearly holding percentage of 7%. The current amount of kanban bins with the current corresponding content are shown in the fourth an fifth column, respectively. This table also shows which component is produced at the manufactured component flow and which component is processed at the buy-in component flow.

| Partnumber | Total Cost per part (euro) | Holding cost (7% per year) | Total no. boxes | Pc. per box | Manufactured (M) or Buy-in (B) |
|---|---|---|---|---|---|
| 401200 | 11.24 | 0.79 | 2 | 100 | B |
| 401201 | 11.24 | 0.79 | 2 | 100 | B |
| 401203 | 11.24 | 0.79 | 2 | 100 | B |
| 401202 | 11.24 | 0.79 | 2 | 100 | B |
| 401131 | 8.37 | 0.59 | 14 | 30 | M |
| 401199 | 8.32 | 0.58 | 2 | 100 | B |
| 401146 | 7.49 | 0.82 | 5 | 30 | M |
| 401145 | 7.29 | 0.51 | 6 | 20 | M |
| 401132 | 7.29 | 0.51 | 18 | 22 | M |
| 401133 | 5.49 | 0.38 | 16 | 25 | M |
| 401052 | 4.73 | 0.33 | 2 | 100 | B |
| 401106 | 4.29 | 0.30 | 2 | 100 | B |
| 401127 | 4.26 | 0.30 | 9 | 60 | M |
| 401108 | 3.75 | 0.26 | 2 | 100 | B |
| 401136 | 2.33 | 0.16 | 6 | 50 | M |
| 401210 | 2.30 | 0.16 | 2 | 100 | B |
| 401053 | 2.24 | 0.16 | 3 | 200 | B |
| 401222 | 2.13 | 0.15 | 5 | 50 | M |
| 401137 | 2.04 | 0.14 | 10 | 50 | M |
| 401134 | 1.51 | 0.11 | 9 | 50 | M |
| 401089 | 1.34 | 0.09 | 2 | 100 | B |
| 401163 | 1.13 | 0.08 | 4 | 50 | B |
| 401111 | 1.11 | 0.08 | 4 | 50 | B |
| 401158 | 1.04 | 0.07 | 2 | 100 | B |
| 401161 | 0.88 | 0.06 | 2 | 100 | B |
| 401105 | 0.85 | 0.06 | 4 | 100 | B |
| 401218 | 0.77 | 0.05 | 4 | 200 | B |
| 401159 | 0.70 | 0.05 | 2 | 100 | B |
| 490115 | 0.59 | 0.04 | 2 | 100 | B |
| 490114 | 0.58 | 0.04 | 2 | 100 | B |
| 401171 | 0.55 | 0.04 | 2 | 100 | B |
| 401220 | 0.51 | 0.04 | 2 | 100 | B |
| 401107 | 0.51 | 0.04 | 2 | 100 | B |
| 401219 | 0.51 | 0.04 | 2 | 100 | B |
| 401221 | 0.51 | 0.04 | 2 | 100 | B |
| 401155 | 0.37 | 0.03 | 3 | 200 | B |
| 401156 | 0.31 | 0.02 | 2 | 100 | B |
| 401055 | 0.30 | 0.02 | 4 | 200 | B |
| 401054 | 0.28 | 0.02 | 4 | 200 | B |
| 401157 | 0.18 | 0.01 | 4 | 200 | B |
| 401215 | 0.06 | 0.0042 | 2 | 100 | B |

Table A1.1: Holding costs and storage boundaries per part

# Appendix A2

## $(Q, s)$ policy

This appendix includes all appendices which are used in the $(Q, s)$ inventory policy calculations are given. First, the complete matlab script to determine the optimal $(Q, s)$ inventory policy is given. Followed by the input file for the matlab script. Finally, the output of the matlab script is shown.

## A2.1 $(Q, s)$ policy matlab script

This appendix provides additional information for Section 5.1.1 and Section 5.1.3. The complete matlab script to determine the optimal $(Q, s)$ inventory policy is given in this appendix. This script determines the optimal reorder levels and order quantities. This script also determines the average amount of full time employees per day needed to handle all orders. The last part of the script includes the calculation for order quantity restrictions.

```
close all
clear all
clc

%% Get input file
[filename, pathname] = uigetfile({'*.xls', '*.xlsx'}, 'Pick an excel
    file');
fullname = fullfile(pathname, filename);

%% Variable
hours_day = xlsread(fullname,'B8:B8') ;
a = xlsread(fullname,'B13:AP13') ; % a is demand per day
stdv_day_type = xlsread(fullname,'B14:AP14');
var_day_type=stdv_day_type.^2;

%% Get values
lt = xlsread(fullname,'B15:AP15') ; % Fixed minimum leadtime if
    ordered
Var_leadtime= xlsread(fullname,'B16:AP16'); % Variable leadtime per
    piece ordered
Ordercost = xlsread(fullname,'B19:AP19'); % Costs ordering one batch
Batchgrootte = xlsread(fullname,'B18:AP18'); % Maximum volume per
    batch
CostProd = xlsread(fullname,'B21:AP21'); % Costs per product
h= xlsread(fullname,'B23:AP23'); % Holding costs per part per day
penaltyp= xlsread(fullname,'B20:AP20'); % Amount of operators effected
     due to shortage
FixedOrdertime =  xlsread(fullname,'B7:B7'); % Fixed ordertime
OperatorCost = xlsread(fullname,'B6:B6'); % Operator costs per hour

%% Calculation

Q = [1:1:2000];
for ii=1:length(a);
```

```
  for jj = 1:length(Q);
      K(ii,jj)= ceil(Q(jj)/Batchgrootte(ii))*Ordercost(ii);

leadtime(ii,jj)= lt(ii) + (Var_leadtime(ii).*Q(jj)) ;
mu(ii,jj)= (leadtime(ii,jj)./hours_day).*a(ii);
sigma(ii,jj) = sqrt((var_day_type(ii).*mu(ii,Q(jj)))./a(ii));
pi(ii,jj)= leadtime(ii,jj).*penaltyp(ii).*OperatorCost ;

  % Bepaal door Cs = 0 de Ks en s waarden
dKs(ii,jj)= 1- (h(ii)*Q(jj))/(pi(ii,jj)*a(ii)) ;   % P stock out
Ks(ii,jj)= norminv(dKs(ii,jj),0,1)   ; % returns inverse cdf, Z value
klfi(ii,jj)= pdf('norm',Ks(ii,jj),0,1);
s(ii,jj) = mu(ii,jj) + (Ks(ii)*sigma(ii,jj))     ;
Gs(ii,jj) = klfi(ii,jj)- (Ks(ii,jj)*(1-dKs(ii,jj)));
Cs(ii,jj) = pi(ii,jj) * sigma(ii,jj) * Gs(ii,jj);

InCost(ii,jj)= h(ii)*((Q(jj)/2)+ s(ii,jj)-mu(ii,jj));
ReCost(ii,jj)= K(ii,jj) * (a(ii) / Q(jj));
StCost(ii,jj)= (a(ii) / Q(jj)) * Cs(ii,jj);
TotalAverageCost(ii,jj) = InCost(ii,jj)+ReCost(ii,jj)+StCost(ii,jj);
  end
end

OptKost = []
for kk=1:length(a);
[M,I] = min(TotalAverageCost((kk),:)) % Get optimal costs and
    corresponding orderquantity per part
OrderGR = ReCost(kk,I)/(Ordercost(kk)*(ceil(I/Batchgrootte(kk)))) %
    Calculate the amount of orders per day
sOpt = ceil(mu(kk,I) + (Ks(kk)*sigma(kk,I)))
ServiceLevel = (1-(Gs(kk,I).*sigma(kk,I)))*100; % Displayed in
    percentages

InCost_opt= h(kk)*((Q(I)/2)+ s(kk,I)-mu(kk,I));
ReCost_opt= K(kk,I) * (a(kk) / Q(I)) ;
StCost_opt= (a(kk) / Q(I)) * Cs(kk,I);
OptKost = [OptKost; M,I,OrderGR,sOpt,InCost_opt ,ReCost_opt,
    StCost_opt, ServiceLevel ]
end

T = array2table(OptKost)
T = array2table(OptKost,...
     'VariableNames',{'OptimalCost','Q','s','InCost_opt','ReCost_opt','
         StCost_opt','AmountOrderPerDay','ServiceLevel'})

PartName = {'401131';'*401146';'*401145';'401132';'401133';'401127';'
    401136';'*401222';'401137';'401134';'A401200';'A401201';'A401203';'
    A401202';'A401199';'401052';'401106';'*401108';'401210';'401053';'
    401089';'401163';'401111';'*401158';'*401161';'401105';'401218';'
    *401159';'*490115';'*490114';'*401171';'401220';'401107';'401219';'
    401221';'401155';'401156';'401055';'401054';'401157';'401215'};
T = table(OptKost(:,1), OptKost(:,2), OptKost(:,4), OptKost(:,5),
    OptKost(:,6),OptKost(:,7), OptKost(:,3),OptKost(:,8) ,...
     'RowNames',PartName)
```

```
T.Properties.VariableNames = {'OptimalCost' 'Q' 's' 'InvCost' 'ReCost' 'StCost' 'AmountOrderPerDay' 'ServiceLevel'}
 uitable('Data',T{:,:},'ColumnName',T.Properties.VariableNames,...
    'RowName',T.Properties.RowNames,'Units', 'Normalized', 'Position',[0, 0, 1, 1]);

%% Calculation of total costs per day per instrument type
OptKostRow= OptKost(:,1);
TotalInventStandard= (FixedOrdertime*OperatorCost)+ OptKostRow(1)+sum(OptKostRow(4:7))+sum(OptKostRow(9:17))+sum(OptKostRow(19:23))+sum(OptKostRow(26:27))+sum(OptKostRow(32:41));
TotalInventSI= (FixedOrdertime*OperatorCost)+ OptKostRow(2)+OptKostRow(3)+ OptKostRow(8)+ OptKostRow(18)+ sum(OptKostRow(24:25))+sum(OptKostRow(28:31));

figure
PartName2 = {'Total Costs per day per product type '};
T2 = table(TotalInventStandard, TotalInventSI,...
    'RowNames',PartName2)
T2.Properties.VariableNames = {'Standard' 'SI' }

uitable('Data',T2{:,:},'ColumnName',T2.Properties.VariableNames,...
    'RowName',T2.Properties.RowNames,'Units', 'Normalized', 'Position',[0, 0, 1, 1]);


%% Calculate FTE, get parameters
M4_time = xlsread(fullname,'B16:AP16')*60; %Get process M4 time manufactured components
Order_time_BOperator = xlsread(fullname,'B34:AP34'); % Get needed B-operator time per part per order
Order_time_AOperator = xlsread(fullname,'B35:AP35'); % Get needed A-operator time per part per order


%% Calculate needed Operator minutes per day
for zz=1:length(a);
M4_time_order(zz)= M4_time(zz)*OptKost(zz,2); % Calculate Needed M4 Time per batch with optimal Q

AmountOrderRow(zz)= OptKost(zz,3) %Get Amount of orders per day per part
AverageOrdertime_B_Operator(zz)= AmountOrderRow(zz)*Order_time_BOperator(zz) + (AmountOrderRow(zz)*M4_time_order(zz)); %B-operator does process M2 an M4
AverageOrdertime_A_Operator(zz)= AmountOrderRow(zz)*Order_time_AOperator(zz);
end

%% Summation of operator minutes per body type per operator type
AverageOrdertime_B_Operator_Standard=(FixedOrdertime*60)+AverageOrdertime_B_Operator(1)+sum(AverageOrdertime_B_Operator(4:7))+sum(AverageOrdertime_B_Operator(9:17))+sum(AverageOrdertime_B_Operator(19:23))+sum(AverageOrdertime_B_Operator
```

```
    (26:27))+sum(AverageOrdertime_B_Operator(32:41));
AverageOrdertime_B_Operator_SI=(FixedOrdertime*60)+
    AverageOrdertime_B_Operator(2)+AverageOrdertime_B_Operator(3)+
    AverageOrdertime_B_Operator(8)+ AverageOrdertime_B_Operator(18)+
    sum(AverageOrdertime_B_Operator(24:25))+sum(
    AverageOrdertime_B_Operator(28:31));
AverageOrdertime_A_Operator_Standard= AverageOrdertime_A_Operator(1)+
    sum(AverageOrdertime_A_Operator(4:7))+sum(
    AverageOrdertime_A_Operator(9:17))+sum(AverageOrdertime_A_Operator
    (19:23))+sum(AverageOrdertime_A_Operator(26:27))+sum(
    AverageOrdertime_A_Operator(32:41));
AverageOrdertime_A_Operator_SI= AverageOrdertime_A_Operator(2)+
    AverageOrdertime_A_Operator(3)+ AverageOrdertime_A_Operator(8)+
    AverageOrdertime_A_Operator(18)+ sum(AverageOrdertime_A_Operator
    (24:25))+sum(AverageOrdertime_A_Operator(28:31));

AverageOrderTime = [AverageOrdertime_B_Operator_Standard
    AverageOrdertime_A_Operator_Standard AverageOrdertime_B_Operator_SI
     AverageOrdertime_A_Operator_SI];
FTE = AverageOrderTime./450;

figure
PartName3 = {'Average operator B time, Standard';'Average operator A
    time, Standard';'Average operator B time, SI';'Average operator A
    time, SI'};
T3 = table(AverageOrderTime', FTE',...
     'RowNames',PartName3)
T3.Properties.VariableNames = {'Minutes' 'FTE' }

uitable('Data',T3{:,:},'ColumnName',T3.Properties.VariableNames,...
     'RowName',T3.Properties.RowNames,'Units', 'Normalized', 'Position'
         ,[0, 0, 1, 1]);
%%
DrawGraph = xlsread(fullname,'B5:B5')
if isempty(DrawGraph)
else
  figure
  scatter(Q,TotalAverageCost(DrawGraph,:))
  title(['Total cost graph of product ' num2str(DrawGraph)])
  xlabel('Order quantity')
  ylabel('Total costs')
end
```

## A2.2 $(Q, s)$ policy input file

The previous appendix showed the complete matlab script to determine the optimal $(Q, s)$ inventory policy. The input file used in the matlab script is shown in this appendix and provides additional information for Section 5.1.1 and Section 5.1.4. As can be seen, many parameters are listed in this input file. The cell 'Graph of product' can be used to establish a graph of the desired product. This graph, shown in Figure 5.1, shows the total policy cost as function of the order quantity. The values 1 to 41 can be filled in which corresponds to the part names given in row 'Partname'. If no value is entered, no graph is created. The yellow marked cells correspond to the demand and standard deviation per day. The row 'Ratio' indicates the ratio of components used per day. For example, components of the standard instrument which are used in every instrument are indicated with 1 and components which are used one out of five times with standard body production are indicated with 0,2. The rows 'Demand per day with ratio' and 'Stdv per day with ratio' are calculated automatically if the demand in the yellow marked cells is filled in. All other parameters can be adjusted when needed. Figure A2.1 shows the first part of the input file and Figure A2.2 the second part.

| Product | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Graph of product | 1 | | | | | | | | | | | | | |
| Operator cost per hour | 32,4 | | | | | | | | | | | | | |
| Fixed ordertime per day [hour] | 0,3333 | | | | | | | | | | | | | |
| Hours a day | 8 | | | | | | | | | | | | | |
| Demand per day | 25,15 | | | | | | | | | | | | | |
| Standard dev day | 3,7 | | | | | | | | | | | | | |
| Part Name | 401131 | 401146 | 401145 | 401132 | 401133 | 401127 | 401136 | 401222 | 401137 | 401134 | A401200 | A401201 | A401203 | A401202 |
| Ratio | 1 | 1 | 1 | 1 | 1 | 1 | 0,2 | 1 | 1 | 0,8 | 0,2 | 0,2 | 0,2 | 0,2 |
| Demand per day with ratio | 25,15 | 25,15 | 25,2 | 25,15 | 25,2 | 25,15 | 5,03 | 25,15 | 25,2 | 20,12 | 5,03 | 5,03 | 5,03 | 5,03 |
| Stdv per day with ratio | 3,7 | 3,7 | 3,7 | 3,7 | 3,7 | 3,7 | 1,65469 | 3,7 | 3,7 | 3,3094 | 1,65469 | 1,65 | 1,65 | 1,65 |
| Fixed Lead time [hours] | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,15 | 2,15 | 2,15 | 2,15 |
| Variable Lead time (inspection) [hours] | 0,0308 | 0,022 | 0,03 | 0,034167 | 0,03 | 0,021 | 0,011111 | 0,01111 | 0,01 | 0,0097 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | |
| BatchSize Bins | 30 | 30 | 20 | 22 | 25 | 60 | 50 | 50 | 50 | 50 | 100 | 100 | 100 | 100 |
| OrderCost per bin | 10,26 | 10,26 | 10,3 | 10,26 | 10,3 | 10,26 | 10,26 | 10,26 | 10,3 | 10,26 | 2,16 | 2,16 | 2,16 | 2,16 |
| Pentalty # amount of people affected with shortage | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 |
| Cost per part | 8,3687 | 7,49 | 7,29 | 7,28865 | 5,49 | 4,255 | 2,3301 | 2,12895 | 2,04 | 1,508 | 11,2388 | 11,2 | 11,2 | 11,2 |
| Holding percentage per day [7%Year] | 0,0002 | 2E-04 | 0 | 0,000192 | 0 | 2E-04 | 0,000192 | 0,00019 | 0 | 0,0002 | 0,00019 | 0 | 0 | 0 |
| Holding costs | 0,0016 | 0,001 | 0 | 0,001398 | 0 | 8E-04 | 0,000447 | 0,00041 | 0 | 0,0003 | 0,00216 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | |
| Operator B time Needed | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 8 | 8 | 8 | 8 |
| Operator A time Needed | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 0 | 0 | 0 | 0 |

Figure A2.1: First part of the input file for the $(Q, s)$ policy script

| Product | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Graph of product | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Operator cost per hour | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fixed ordertime per day [hour] | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hours a day | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Demand per day | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Standard dev day | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Part Name | A401199 | 401052 | 401106 | 401108 | 401210 | 401053 | 401089 | 401136 | 401111 | 401158 | 401161 | 401105 | 401218 | 401159 | 490115 | 490114 | 401171 | 401220 | 401107 | 401219 | 401221 | 401155 | 401156 | 401055 | 401054 | 401157 | 401215 |
| Ratio | 0,2 | 0,8 | 0,8 | 1 | 0,2 | 0,8 | 0,2 | 0,2 | 0,8 | 1 | 1 | 0,8 | 1 | 1 | 1 | 1 | 1 | 0,2 | 0,2 | 0,2 | 0,2 | 0,8 | 0,2 | 1 | 1 | 1 | 0,2 |
| Demand per day with ratio | 5,03 | 20,1 | 20,1 | 25,2 | 5,03 | 20,1 | 5,03 | 5,03 | 20,1 | 25,2 | 25,2 | 20,1 | 25,2 | 25,2 | 25,2 | 25,2 | 25,2 | 5,03 | 5,03 | 5,03 | 5,03 | 20,1 | 5,03 | 25,2 | 25,2 | 25,2 | 5,03 |
| Stdv per day with ratio | 1,65 | 3,31 | 3,31 | 3,7 | 1,65 | 3,31 | 1,65 | 1,65 | 3,31 | 3,7 | 3,7 | 3,31 | 3,7 | 3,7 | 3,7 | 3,7 | 3,7 | 1,65 | 1,65 | 1,65 | 1,65 | 3,31 | 1,65 | 3,7 | 3,7 | 3,7 | 1,655 |
| Fixed Lead time [hours] | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 |
| Variable Lead time (inspection) [hours] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BatchSize Bins | 100 | 100 | 100 | 100 | 100 | 200 | 100 | 50 | 50 | 100 | 100 | 100 | 200 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 200 | 100 | 200 | 200 | 200 | 100 |
| OrderCost per bin | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 |
| Pentalty # amount of people affected with shortage | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Cost per part | 8,32 | 4,73 | 4,29 | 3,75 | 2,3 | 2,24 | 1,34 | 1,13 | 1,11 | 1,04 | 0,88 | 0,85 | 0,77 | 0,7 | 0,59 | 0,58 | 0,55 | 0,51 | 0,51 | 0,51 | 0,51 | 0,37 | 0,31 | 0,3 | 0,28 | 0,18 | 0,06 |
| Holding percentage per day [7%Year] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2E-04 |
| Holding costs | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1E-05 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Operator B time Needed | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Operator A time Needed | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure A2.2: Second part of the input file for the $(Q, s)$ policy script

## A2.3 $(Q, s)$ policy output

This appendix provides additional information for Section 5.1.1. Figure A2.3 shows the output table created by the $(Q, s)$ policy matlab script for all components. The first column displays the component numbers. Numbers starting with a star are used for suction irrigation devices. All other components are used in the standard instruments. The second column shows the total average cost per day per component. The third and fourth column show the optimal order quantity and optimal reorder level respectively. Column five until seven show the average inventory cost, average replenishment cost and average shortage cost, respectively. The eighth column shows the average amount of orders per day for the corresponding part. The last column gives the service levels per component.

| | Optimal... | Q | s | InvCost | ReCost | StCost | AmountOrderP... | Service... |
|---|---|---|---|---|---|---|---|---|
| 401131 | 0.9593 | 554 | 92 | 0.4915 | 0.4658 | 0.0021 | 0.0454 | 99.9976 |
| *401146 | 0.9004 | 588 | 76 | 0.4599 | 0.4388 | 0.0017 | 0.0428 | 99.9975 |
| *401145 | 0.8956 | 593 | 102 | 0.4586 | 0.4351 | 0.0019 | 0.0424 | 99.9980 |
| 401132 | 0.8958 | 593 | 104 | 0.4587 | 0.4351 | 0.0019 | 0.0424 | 99.9980 |
| 401133 | 0.7717 | 686 | 103 | 0.3941 | 0.3762 | 0.0014 | 0.0367 | 99.9982 |
| 401127 | 0.6739 | 782 | 89 | 0.3429 | 0.3300 | 0.0010 | 0.0322 | 99.9983 |
| 401136 | 0.2186 | 478 | 14 | 0.1105 | 0.1080 | 1.7108e-04 | 0.0105 | 99.9979 |
| *401222 | 0.4704 | 1113 | 75 | 0.2382 | 0.2318 | 4.4682e-04 | 0.0226 | 99.9986 |
| 401137 | 0.4586 | 1140 | 58 | 0.2319 | 0.2263 | 3.6954e-04 | 0.0221 | 99.9984 |
| 401134 | 0.3526 | 1185 | 60 | 0.1781 | 0.1742 | 2.7497e-04 | 0.0170 | 99.9988 |
| A401200 | 0.2252 | 100 | 6 | 0.1161 | 0.1086 | 4.7283e-04 | 0.0503 | 99.9933 |
| A401201 | 0.2252 | 100 | 6 | 0.1161 | 0.1086 | 4.7283e-04 | 0.0503 | 99.9933 |
| A401203 | 0.2252 | 100 | 6 | 0.1161 | 0.1086 | 4.7283e-04 | 0.0503 | 99.9933 |
| A401202 | 0.2252 | 100 | 6 | 0.1161 | 0.1086 | 4.7283e-04 | 0.0503 | 99.9933 |
| A401199 | 0.1929 | 117 | 6 | 0.0997 | 0.0929 | 3.4706e-04 | 0.0430 | 99.9942 |
| 401052 | 0.2888 | 310 | 14 | 0.1482 | 0.1402 | 3.7232e-04 | 0.0649 | 99.9959 |
| 401106 | 0.2747 | 325 | 14 | 0.1407 | 0.1337 | 3.3694e-04 | 0.0619 | 99.9961 |
| *401108 | 0.2868 | 389 | 17 | 0.1468 | 0.1397 | 3.2627e-04 | 0.0647 | 99.9964 |
| 401210 | 0.0997 | 222 | 6 | 0.0507 | 0.0489 | 9.2107e-05 | 0.0227 | 99.9971 |
| 401053 | 0.1972 | 450 | 15 | 0.1005 | 0.0966 | 1.7307e-04 | 0.0447 | 99.9972 |
| 401089 | 0.0759 | 291 | 6 | 0.0385 | 0.0373 | 5.2984e-05 | 0.0173 | 99.9978 |
| 401163 | 0.0697 | 316 | 6 | 0.0353 | 0.0344 | 4.4621e-05 | 0.0159 | 99.9980 |
| 401111 | 0.1382 | 638 | 15 | 0.0700 | 0.0681 | 8.4354e-05 | 0.0315 | 99.9981 |
| *401158 | 0.1496 | 737 | 17 | 0.0758 | 0.0737 | 8.7826e-05 | 0.0341 | 99.9982 |
| *401161 | 0.1372 | 802 | 17 | 0.0694 | 0.0677 | 7.3747e-05 | 0.0314 | 99.9983 |
| 401105 | 0.1209 | 728 | 15 | 0.0612 | 0.0597 | 6.4403e-05 | 0.0276 | 99.9983 |
| 401218 | 0.1284 | 857 | 18 | 0.0650 | 0.0634 | 6.4511e-05 | 0.0293 | 99.9984 |
| *401159 | 0.1223 | 899 | 18 | 0.0618 | 0.0604 | 5.8415e-05 | 0.0280 | 99.9985 |
| *490115 | 0.1125 | 977 | 18 | 0.0568 | 0.0556 | 4.9304e-05 | 0.0257 | 99.9986 |
| *490114 | 0.1112 | 988 | 18 | 0.0562 | 0.0550 | 4.8156e-05 | 0.0255 | 99.9986 |
| *401171 | 0.1081 | 1016 | 18 | 0.0546 | 0.0535 | 4.5482e-05 | 0.0248 | 99.9987 |
| 401220 | 0.0466 | 471 | 6 | 0.0235 | 0.0231 | 1.9661e-05 | 0.0107 | 99.9987 |
| 401107 | 0.0466 | 471 | 6 | 0.0235 | 0.0231 | 1.9661e-05 | 0.0107 | 99.9987 |
| 401219 | 0.0466 | 471 | 6 | 0.0235 | 0.0231 | 1.9661e-05 | 0.0107 | 99.9987 |
| 401221 | 0.0466 | 471 | 6 | 0.0235 | 0.0231 | 1.9661e-05 | 0.0107 | 99.9987 |
| 401155 | 0.0793 | 1106 | 15 | 0.0400 | 0.0393 | 2.7349e-05 | 0.0182 | 99.9989 |
| 401156 | 0.0363 | 603 | 6 | 0.0183 | 0.0180 | 1.1833e-05 | 0.0083 | 99.9990 |
| 401055 | 0.0791 | 1384 | 18 | 0.0399 | 0.0393 | 2.4142e-05 | 0.0182 | 99.9990 |
| 401054 | 0.0775 | 1414 | 18 | 0.0390 | 0.0384 | 2.3116e-05 | 0.0178 | 99.9991 |
| 401157 | 0.0613 | 1783 | 18 | 0.0308 | 0.0305 | 1.4375e-05 | 0.0141 | 99.9993 |
| 401215 | 0.0158 | 1376 | 7 | 0.0079 | 0.0079 | 2.1835e-06 | 0.0037 | 99.9996 |

Figure A2.3: Output of matlab script for $(Q, s)$ policy

# Appendix A3

## $(R, S)$ policy

This appendix includes all appendices used in the $(R, S)$ inventory policy calculations are given. First, the complete matlab script to determine the optimal $(R, S)$ inventory policy is given. Followed by the input file for the matlab script. Finally, the output of the matlab script is shown.

## A3.1 $(R, S)$ policy matlab script

The complete matlab script to determine the optimal $(R, S)$ inventory policy is given in this appendix. This appendix provides additional information for Section 5.2.1, Section 5.2.2 and Section 5.2.3. This script determines the optimal order levels and the optimal review period. This script also determines the average amount of full time employees per day needed to handle all orders. The last part of the script includes the calculation for order quantity restrictions.

```
close all
clear all
clc

[filename, pathname] = uigetfile({'*.xls', '*.xlsx'}, 'Pick an excel
    file');
fullname = fullfile(pathname, filename);

%% Variable
Fixed_R = xlsread(fullname,'B4:B4');
hours_day = xlsread(fullname,'B8:B8') ;
a = xlsread(fullname,'B13:AP13') ; % a is demand per day
stdv_day_type = xlsread(fullname,'B14:AP14');
var_day_type=stdv_day_type.^2;

%% Get values
lt = xlsread(fullname,'B15:AP15') ; % Fixed minimum leadtime if
    ordered
Var_leadtime= xlsread(fullname,'B16:AP16'); % Variable leadtime per
    piece ordered
Ordercost = xlsread(fullname,'B19:AP19'); % Costs ordering one batch
Batchgrootte = xlsread(fullname,'B18:AP18'); % Maximum volume per
    batch
CostProd = xlsread(fullname,'B21:AP21'); % Costs per product
h= xlsread(fullname,'B23:AP23'); % Holding costs per part per day
penaltyp= xlsread(fullname,'B20:AP20'); % Amount of operators effected
     due to shortage
FixedOrdertime =  xlsread(fullname,'B7:B7'); % Fixed ordertime
OperatorCost = xlsread(fullname,'B6:B6'); % Operator costs per hour
OptKost = []
%% Calculation
int=1;
if isempty(Fixed_R)
   R = [1:int:300];
 else
```

```
  R = Fixed_R
end

for jj = 1:length(R);
for ii=1:length(a);

leadtime(jj,ii)= lt(ii) + (Var_leadtime(ii).*Batchgrootte(ii)) ;
mu(jj,ii)= (R(jj)+(leadtime(ii)./hours_day)).*a(ii);
sigma(jj,ii)= stdv_day_type(ii)*sqrt(R(jj)+(leadtime(jj,ii)./hours_day));
pi(jj,ii)= leadtime(ii).*penaltyp(ii).*OperatorCost ;

% Calulate probability of shortage
dKs(jj,ii)= 1- ((h(ii).*R(jj))/(pi(jj,ii))) ; % Kans goed grijpen
Ks(jj,ii)= norminv(dKs(jj,ii),0,1) ;  % bijbehoorde z waarde
klfi(jj,ii)= pdf('norm',Ks(jj,ii),0,1);
Gs(jj,ii) = klfi(jj,ii)- (Ks(jj,ii)*(1-dKs(jj,ii)));

% Calculate reorder level and safety stock
S(jj,ii) = (mu(jj,ii)+(sigma(jj,ii)*Ks(jj,ii)));
Cs(jj,ii) = pi(jj,ii).*(1 - dKs(jj,ii)); %pi*[1-F(S)]
SS(jj,ii) = S(jj,ii)-mu(jj,ii);

K(jj,ii)= ceil((((S(jj,ii)-SS(jj,ii)))/Batchgrootte(ii)))*Ordercost(ii); %Tot order costs per R

InCost(jj,ii)= h(ii)*((((a(ii).*R(jj))/2)+ S(jj,ii)-mu(jj,ii))); % InCost = h(ii)*(((a*R)/2)+ S-mu));
ReCost(jj,ii)= K(jj,ii)./R(jj) ; % ReCost= (K/R);
StCost(jj,ii)= (1./R(jj)).*Cs(jj,ii); % StCost= (1/R)*Cs;
TotalAverageCost(jj,ii) = InCost(jj,ii)+ReCost(jj,ii)+StCost(jj,ii);
     end

end

Averagemu=[];
Rperiod=[] ;
        for ii=1:length(a);
[M,I] = min(TotalAverageCost(:,ii)) % Get minimal costs with corresponding R
if isempty(Fixed_R)
    Opt = M;
    Rev = I * int;
 else
    Opt = M;
    Rev = Fixed_R ;
end

SOpt = ceil(mu(I,ii) + (Ks(I,ii)*sigma(I,ii))); % optimal reorder level

SS2 = ceil(SOpt - mu(I,ii));
ServiceLevel = (1-(Gs(I,ii).*sigma(I,ii)))*100;% Displayed in percentages
```

```
Batchgr = Batchgrootte(ii);

InCost_opt= h(ii)*(((a(ii).*Rev)/2)+ S(I,ii)-mu(I,ii)); % InCost = h(
    ii)*(((a*R)/2)+ S-mu));
ReCost_opt= K(I,ii)./Rev ; % ReCost= (K/R);
StCost_opt = (1./Rev).*Cs(I,ii); % StCost= (1/R)*Cs4;

avermu = mu(I,ii);
OrderGR = ceil(mu(I,ii)/Batchgrootte(ii));
AmountDay= OrderGR/Rev;

 OptKost = [OptKost; M,SOpt,SS2,InCost_opt ,ReCost_opt, StCost_opt,
     AmountDay,ServiceLevel,Rev, OrderGR];
 Averagemu= [Averagemu; avermu]
 Rperiod = [Rperiod; Rev]
      end

 figure
 scatter(R,TotalAverageCost(:,1))


T = array2table(OptKost)
T = array2table(OptKost,...
     'VariableNames',{'OptimalCost','S','SS','InCost_opt','ReCost_opt',
         'StCost_opt','AmountOrderPerDay','ServiceLevel','
         Opt_ReorderPeriod','OrderGR'})

PartName = {'401131';'*401146';'*401145';'401132';'401133';'401127';'
    401136';'*401222';'401137';'401134';'A401200';'A401201';'A401203';'
    A401202';'A401199';'401052';'401106';'*401108';'401210';'401053';'
    401089';'401163';'401111';'*401158';'*401161';'401105';'401218';'
    *401159';'*490115';'*490114';'*401171';'401220';'401107';'401219';'
    401221';'401155';'401156';'401055';'401054';'401157';'401215'};
T = table(OptKost(:,1), OptKost(:,2), OptKost(:,3), OptKost(:,4),
    OptKost(:,5),OptKost(:,6),OptKost(:,7),OptKost(:,8),OptKost(:,9),
    OptKost(:,10) ,...
     'RowNames',PartName)
T.Properties.VariableNames = {'OptimalCost' 'S' 'SS' 'InvCost' 'ReCost
    ' 'StCost' 'AVGAmountOrderPerDay' 'ServiceLevel' 'Opt_ReorderPeriod
    ' 'NumberBins' }
 uitable('Data',T{:,:},'ColumnName',T.Properties.VariableNames,...
     'RowName',T.Properties.RowNames, 'Units', 'Normalized', 'Position'
         ,[0, 0, 1, 1]);

%% Calculation of total costs per day per instrument type
OptKostRow= OptKost(:,1);
TotalInventStandard= (FixedOrdertime*OperatorCost)+ OptKostRow(1)+sum(
    OptKostRow(4:7))+sum(OptKostRow(9:17))+sum(OptKostRow(19:23))+sum(
    OptKostRow(26:27))+sum(OptKostRow(32:41));
TotalInventSI= (FixedOrdertime*OperatorCost)+ OptKostRow(2)+OptKostRow
    (3)+ OptKostRow(8)+ OptKostRow(18)+ sum(OptKostRow(24:25))+sum(
    OptKostRow(28:31));

figure
```

```
PartName2 = {'Total Costs per day per product type '};
T2 = table(TotalInventStandard, TotalInventSI,...
    'RowNames',PartName2)
T2.Properties.VariableNames = {'Standard' 'SI' }

uitable('Data',T2{:,:},'ColumnName',T2.Properties.VariableNames,...
    'RowName',T2.Properties.RowNames,'Units', 'Normalized', 'Position',[0, 0, 1, 1]);
%% Calculate FTE, get parameters
M4_time = xlsread(fullname,'B16:AP16')*60; %Get process M4 time manufactured components
Order_time_BOperator = xlsread(fullname,'B34:AP34'); % Get needed B-operator time per part per order
Order_time_AOperator = xlsread(fullname,'B35:AP35'); % Get needed A-operator time per part per order


%% Calculate needed Operator minutes per day
for zz=1:length(a);
M4_time_order(zz)= M4_time(zz).*(Averagemu(zz)./Rperiod(zz)) ;% Calculate Needed M4 Time per day
AmountOrderRow(zz)= OptKost(zz,7) %Get Amount of orders per day per part
AverageOrdertime_B_Operator(zz)= AmountOrderRow(zz)*Order_time_BOperator(zz) + (M4_time_order(zz)); %B-operator does M2 and M4
AverageOrdertime_A_Operator(zz)= AmountOrderRow(zz)*Order_time_AOperator(zz);
end

%% Summation of operator minutes per body type per operator type
AverageOrdertime_B_Operator_Standard=(FixedOrdertime*60)+AverageOrdertime_B_Operator(1)+sum(AverageOrdertime_B_Operator(4:7))+sum(AverageOrdertime_B_Operator(9:17))+sum(AverageOrdertime_B_Operator(19:23))+sum(AverageOrdertime_B_Operator(26:27))+sum(AverageOrdertime_B_Operator(32:41));
AverageOrdertime_B_Operator_SI=(FixedOrdertime*60)+AverageOrdertime_B_Operator(2)+AverageOrdertime_B_Operator(3)+AverageOrdertime_B_Operator(8)+ AverageOrdertime_B_Operator(18)+sum(AverageOrdertime_B_Operator(24:25))+sum(AverageOrdertime_B_Operator(28:31));
AverageOrdertime_A_Operator_Standard= AverageOrdertime_A_Operator(1)+sum(AverageOrdertime_A_Operator(4:7))+sum(AverageOrdertime_A_Operator(9:17))+sum(AverageOrdertime_A_Operator(19:23))+sum(AverageOrdertime_A_Operator(26:27))+sum(AverageOrdertime_A_Operator(32:41));
AverageOrdertime_A_Operator_SI= AverageOrdertime_A_Operator(2)+AverageOrdertime_A_Operator(3)+ AverageOrdertime_A_Operator(8)+AverageOrdertime_A_Operator(18)+ sum(AverageOrdertime_A_Operator(24:25))+sum(AverageOrdertime_A_Operator(28:31));

AverageOrderTime = [AverageOrdertime_B_Operator_Standard AverageOrdertime_A_Operator_Standard AverageOrdertime_B_Operator_SI AverageOrdertime_A_Operator_SI];
```

```
FTE = AverageOrderTime./450;

figure
PartName3 = {'Average operator B time, Standard';'Average operator A time, Standard';'Average operator B time, SI';'Average operator A time, SI'};
T3 = table(AverageOrderTime', FTE',...
    'RowNames',PartName3)
T3.Properties.VariableNames = {'Minutes' 'FTE' }

uitable('Data',T3{:,:},'ColumnName',T3.Properties.VariableNames,...
    'RowName',T3.Properties.RowNames,'Units', 'Normalized', 'Position',[0, 0, 1, 1]);

%

DrawGraph = xlsread(fullname,'B5:B5')
if isempty(DrawGraph)
  % do something
else
  figure
  scatter(R,TotalAverageCost(:,DrawGraph))
  title(['Total cost graph of product ' num2str(DrawGraph)])
  xlabel('Review period')
  ylabel('Total costs')
end
```

## A3.2 $(R, S)$ policy input file

The previous appendix showed the complete matlab script to determine the optimal $(R, S)$ inventory policy. The input file used in the $(R, S)$ policy matlab script is shown in this appendix and provides additional information for Section 5.2.1. As can be seen, many parameters are listed in this input file. The first cell 'Review Period' is used to choose whether the calculation is done using a fixed review period or a review period range. If the cell is empty, a range of review periods is used to calculated the optimal $(R, S)$ inventory policy. If a value is entered, it is used as fixed review period. The cell 'Graph of product' can be used to establish a graph of the desired product. This graph, shown in Figure A3.1, shows the total policy cost as function of the review period. The values 1 to 41 can be filled in which corresponds to the part names given in row 'Partname'. If no value is entered, no graph is created. The yellow marked cells correspond to the demand and standard deviation per day. The row 'Ratio' indicates the ratio of components used per day. The rows 'Demand per day with ratio' and 'Stdv per day with ratio' are calculated automatically if the demand in the yellow marked cells is filled in. All other parameters can be adjusted when needed. Figure A3.2 shows the first part of the input file and Figure A3.3 the second part.

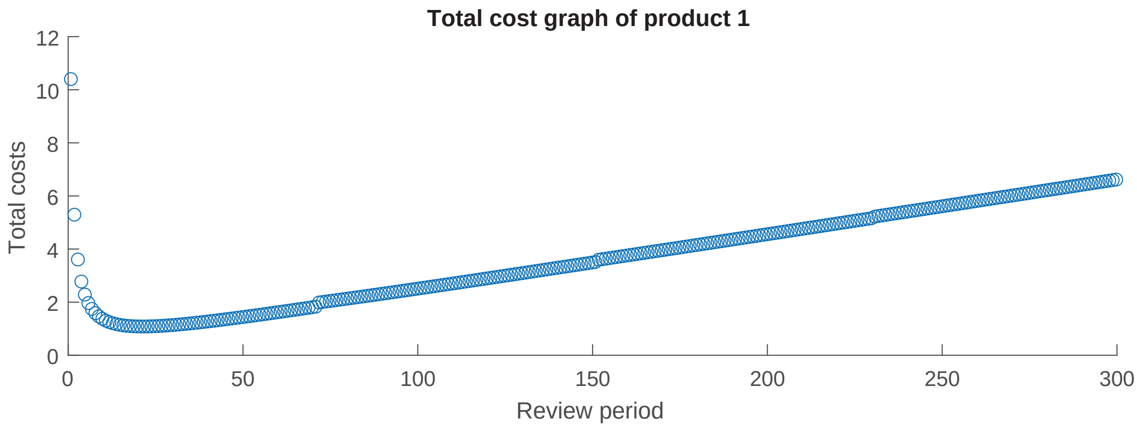


Figure A3.1: Graph of total $(R, S)$ policy cost as function of the review period for product 1, which corresponds to part number 401131.

| Product | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review Period [days] | | | | | | | | | | | | | | |
| Graph of product | 1 | | | | | | | | | | | | | |
| Operator cost per hour | 32,4 | | | | | | | | | | | | | |
| Fixed ordertime per day [hour] | 0,33333 | | | | | | | | | | | | | |
| Hours a day | 8 | | | | | | | | | | | | | |
| Demand per day | 25,15 | | | | | | | | | | | | | |
| Standard dev day | 3,7 | | | | | | | | | | | | | |
| Part Name | 401131 | 401146 | 401145 | 401132 | 401133 | 401127 | 401136 | 401222 | 401137 | 401134 | A401200 | A401201 | A401203 | A401202 |
| Ratio | 1 | 1 | 1 | 1 | 1 | 1 | 0,2 | 1 | 1 | 0,8 | 0,2 | 0,2 | 0,2 | 0,2 |
| Demand per day with ratio | 25,15 | 25,2 | 25,2 | 25,15 | 25,2 | 25,15 | 5,03 | 25,15 | 25,2 | 20,12 | 5,03 | 5,03 | 5,03 | 5,03 |
| Stdv per day with ratio | 3,7 | 3,7 | 3,7 | 3,7 | 3,7 | 3,7 | 1,65469 | 3,7 | 3,7 | 3,3094 | 1,65469 | 1,65 | 1,65 | 1,65 |
| Fixed Lead time [hours] | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,65 | 2,15 | 2,15 | 2,15 | 2,15 |
| Variable Lead time (inspection) [hours] | 0,03083 | 0,02 | 0,03 | 0,03417 | 0,03 | 0,021 | 0,011111 | 0,01111 | 0,01 | 0,0097 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | |
| BatchSize Bins | 30 | 30 | 20 | 22 | 25 | 60 | 50 | 50 | 50 | 50 | 100 | 100 | 100 | 100 |
| OrderCost per bin | 10,26 | 10,3 | 10,3 | 10,26 | 10,3 | 10,26 | 10,26 | 10,26 | 10,3 | 10,26 | 2,16 | 2,16 | 2,16 | 2,16 |
| Pentalty # amount of people affected with shortage | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 |
| Cost per part | 8,36865 | 7,49 | 7,29 | 7,28865 | 5,49 | 4,255 | 2,3301 | 2,12895 | 2,04 | 1,508 | 11,2388 | 11,2 | 11,2 | 11,2 |
| Holding percentage per day [7%Year] | 0,00019 | 0 | 0 | 0,00019 | 0 | 2E-04 | 0,000192 | 0,00019 | 0 | 0,0002 | 0,00019 | 0 | 0 | 0 |
| Holding costs | 0,0016 | 0 | 0 | 0,0014 | 0 | 8E-04 | 0,000447 | 0,00041 | 0 | 0,0003 | 0,00216 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | |
| Operator B Time Needed | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 8 | 8 | 8 | 8 |
| Operator A Time Needed | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 0 | 0 | 0 | 0 |

Figure A3.2: First part of the input file for the $(R, S)$ policy script

| Product | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review Period [days] | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Graph of product | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Operator cost per hour | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fixed ordertime per day [hour] | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hours a day | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Demand per day | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Standard dev day | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **Part Name** | A401199 | 401052 | 401106 | 401108 | 401210 | 401053 | 401089 | 401136 | 401111 | 401158 | 401161 | 401105 | 401218 | 401159 | 490115 | 490114 | 401171 | 401220 | 401107 | 401219 | 401221 | 401155 | 401156 | 401055 | 401054 | 401157 | 401215 |
| Ratio | 0,2 | 0,8 | 0,8 | 1 | 0,2 | 0,8 | 0,2 | 0,2 | 0,8 | 1 | 1 | 0,8 | 1 | 1 | 1 | 1 | 1 | 0,2 | 0,2 | 0,2 | 0,2 | 0,8 | 0,2 | 1 | 1 | 1 | 0,2 |
| Demand per day with ratio | 5,03 | 20,1 | 20,1 | 25,2 | 5,03 | 20,1 | 5,03 | 5,03 | 20,1 | 25,2 | 25,2 | 20,1 | 25,2 | 25,2 | 25,2 | 25,2 | 25,2 | 5,03 | 5,03 | 5,03 | 5,03 | 20,1 | 5,03 | 25,2 | 25,2 | 25,2 | 5,03 |
| Stdv per day with ratio | 1,65 | 3,31 | 3,31 | 3,7 | 1,65 | 3,31 | 1,65 | 1,65 | 3,31 | 3,7 | 3,7 | 3,31 | 3,7 | 3,7 | 3,7 | 3,7 | 3,7 | 1,65 | 1,65 | 1,65 | 1,65 | 3,31 | 1,65 | 3,7 | 3,7 | 3,7 | 1,655 |
| Fixed Lead time [hours] | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 | 2,15 |
| Variable Lead time (inspection) [hours] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BatchSize Bins | 100 | 100 | 100 | 100 | 100 | 200 | 100 | 50 | 50 | 100 | 100 | 100 | 200 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 200 | 100 | 200 | 200 | 200 | 100 |
| OrderCost per bin | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 | 2,16 |
| Pentalty # amount of people affected with shortage | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Cost per part | 8,32 | 4,73 | 4,29 | 3,75 | 2,3 | 2,24 | 1,34 | 1,13 | 1,11 | 1,04 | 0,88 | 0,85 | 0,77 | 0,7 | 0,59 | 0,58 | 0,55 | 0,51 | 0,51 | 0,51 | 0,51 | 0,37 | 0,31 | 0,3 | 0,28 | 0,18 | 0,06 |
| Holding percentage per day [7%Year] | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2E-04 |
| Holding costs | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1E-05 |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Operator B Time Needed | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Operator A Time Needed | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure A3.3: Second part of the input file for the $(R, S)$ policy script

## A3.3 $(R, S)$ policy output

This appendix provides additional information for Section 5.2.1. Figure A3.4 shows the output table created by the $(R, S)$ policy matlab script for all components. The first column displays the component numbers. Numbers starting with a star are used for suction irrigation devices. All other components are used in the standard instruments. The second column shows the total average cost per day per component. The third and fourth column show the optimal order level and the corresponding safety stock. Column five until seven show the average inventory cost, average replenishment cost and average shortage cost, respectively. The eighth column shows the average amount of orders per day for the corresponding component and the ninth column gives the service levels per component. The tenth column shows the optimal review period in days and the last column shows the amount of bins ordered per at this review period.

| | Optimal... | S | SS | InvCost | ReCost | StCost | AVGAmountOrderPe... | Service... | Opt_Reorder... | NumberBins |
|---|---|---|---|---|---|---|---|---|---|---|
| 401131 | 1.0548 | 819 | 89 | 0.5647 | 0.4886 | 0.0016 | 0.0476 | 99.9978 | 21 | 1 |
| *401146 | 0.9888 | 869 | 89 | 0.5413 | 0.4461 | 0.0014 | 0.0435 | 99.9978 | 23 | 1 |
| *401145 | 0.9805 | 873 | 93 | 0.5331 | 0.4461 | 0.0014 | 0.0435 | 99.9978 | 23 | 1 |
| 401132 | 0.9806 | 873 | 93 | 0.5331 | 0.4461 | 0.0014 | 0.0435 | 99.9978 | 23 | 1 |
| 401133 | 0.8397 | 978 | 97 | 0.4587 | 0.3800 | 0.0011 | 0.0370 | 99.9980 | 27 | 1 |
| 401127 | 0.7311 | 1056 | 100 | 0.3882 | 0.3420 | 8.1607e-04 | 0.0333 | 99.9982 | 30 | 1 |
| 401136 | 0.2467 | 563 | 70 | 0.1323 | 0.1140 | 4.4687e-04 | 0.0111 | 99.9978 | 90 | 1 |
| *401222 | 0.5064 | 1398 | 115 | 0.2673 | 0.2386 | 4.0829e-04 | 0.0233 | 99.9986 | 43 | 1 |
| 401137 | 0.4947 | 1424 | 116 | 0.2611 | 0.2332 | 3.9120e-04 | 0.0227 | 99.9986 | 44 | 1 |
| 401134 | 0.3798 | 1426 | 118 | 0.1995 | 0.1800 | 2.8920e-04 | 0.0175 | 99.9986 | 57 | 1 |
| A401200 | 0.2850 | 162 | 32 | 0.1628 | 0.1200 | 0.0022 | 0.0556 | 99.9986 | 18 | 1 |
| A401201 | 0.2850 | 162 | 32 | 0.1628 | 0.1200 | 0.0022 | 0.0556 | 99.9986 | 18 | 1 |
| A401203 | 0.2850 | 162 | 32 | 0.1628 | 0.1200 | 0.0022 | 0.0556 | 99.9986 | 18 | 1 |
| A401202 | 0.2850 | 162 | 32 | 0.1628 | 0.1200 | 0.0022 | 0.0556 | 99.9986 | 18 | 1 |
| A401199 | 0.2413 | 179 | 33 | 0.1368 | 0.1029 | 0.0016 | 0.0476 | 99.9987 | 21 | 1 |
| 401052 | 0.3333 | 456 | 56 | 0.1781 | 0.1543 | 9.0616e-04 | 0.0714 | 99.9989 | 14 | 1 |
| 401106 | 0.3164 | 478 | 58 | 0.1715 | 0.1440 | 8.2231e-04 | 0.0667 | 99.9989 | 15 | 1 |
| *401108 | 0.3269 | 564 | 64 | 0.1719 | 0.1543 | 7.1908e-04 | 0.0714 | 99.9991 | 14 | 1 |
| 401210 | 0.1193 | 289 | 48 | 0.0649 | 0.0540 | 4.4014e-04 | 0.0250 | 99.9991 | 40 | 1 |
| 401053 | 0.2242 | 655 | 71 | 0.1209 | 0.1029 | 4.2978e-04 | 0.0476 | 99.9993 | 21 | 1 |
| 401089 | 0.0893 | 362 | 55 | 0.0483 | 0.0408 | 2.5701e-04 | 0.0189 | 99.9992 | 53 | 1 |
| 401163 | 0.0816 | 390 | 58 | 0.0441 | 0.0372 | 2.1746e-04 | 0.0172 | 99.9992 | 58 | 1 |
| 401111 | 0.1547 | 851 | 86 | 0.0825 | 0.0720 | 2.1333e-04 | 0.0333 | 99.9994 | 30 | 1 |
| *401158 | 0.1663 | 999 | 93 | 0.0889 | 0.0771 | 2.0011e-04 | 0.0357 | 99.9995 | 28 | 1 |
| *401161 | 0.1520 | 1053 | 97 | 0.0799 | 0.0720 | 1.6875e-04 | 0.0333 | 99.9995 | 30 | 1 |
| 401105 | 0.1347 | 937 | 92 | 0.0710 | 0.0635 | 1.6396e-04 | 0.0294 | 99.9995 | 34 | 1 |
| 401218 | 0.1420 | 1107 | 101 | 0.0743 | 0.0675 | 1.4809e-04 | 0.0313 | 99.9995 | 32 | 1 |
| *401159 | 0.1349 | 1160 | 103 | 0.0713 | 0.0635 | 1.3442e-04 | 0.0294 | 99.9995 | 34 | 1 |
| *490115 | 0.1237 | 1241 | 109 | 0.0652 | 0.0584 | 1.1392e-04 | 0.0270 | 99.9996 | 37 | 1 |
| *490114 | 0.1222 | 1241 | 109 | 0.0638 | 0.0584 | 1.1133e-04 | 0.0270 | 99.9996 | 37 | 1 |
| *401171 | 0.1187 | 1267 | 110 | 0.0618 | 0.0568 | 1.0529e-04 | 0.0263 | 99.9996 | 38 | 1 |
| 401220 | 0.0533 | 550 | 72 | 0.0284 | 0.0248 | 9.7892e-05 | 0.0115 | 99.9994 | 87 | 1 |
| 401107 | 0.0533 | 550 | 72 | 0.0284 | 0.0248 | 9.7892e-05 | 0.0115 | 99.9994 | 87 | 1 |
| 401219 | 0.0533 | 550 | 72 | 0.0284 | 0.0248 | 9.7892e-05 | 0.0115 | 99.9994 | 87 | 1 |
| 401221 | 0.0533 | 550 | 72 | 0.0284 | 0.0248 | 9.7892e-05 | 0.0115 | 99.9994 | 87 | 1 |
| 401155 | 0.0869 | 1323 | 116 | 0.0453 | 0.0415 | 7.1080e-05 | 0.0192 | 99.9996 | 52 | 1 |
| 401156 | 0.0410 | 691 | 83 | 0.0219 | 0.0191 | 5.9680e-05 | 0.0088 | 99.9994 | 113 | 1 |
| 401055 | 0.0860 | 1666 | 131 | 0.0452 | 0.0408 | 5.6726e-05 | 0.0189 | 99.9996 | 53 | 1 |
| 401054 | 0.0841 | 1692 | 132 | 0.0441 | 0.0400 | 5.4370e-05 | 0.0185 | 99.9996 | 54 | 1 |
| 401157 | 0.0661 | 2062 | 150 | 0.0343 | 0.0318 | 3.4179e-05 | 0.0147 | 99.9997 | 68 | 1 |
| 401215 | 0.0173 | 1488 | 130 | 0.0090 | 0.0082 | 1.1469e-05 | 0.0038 | 99.9996 | 262 | 1 |

Figure A3.4: Output of matlab script for $(R, S)$policy

# Appendix A4

## Simevents model

This appendix includes all models created in Simevents. First the implementation of operators in the total assembly line is given.

## A4.1 Model with operators

The implementation of operators in the total assembly line simulation is shown in this appendix. This appendix provides additional information for Section 6.3. The model consists of four resource pools with operators which are used at the workstations. As can be seen, process A2.2 and process A5 do not require an operator.

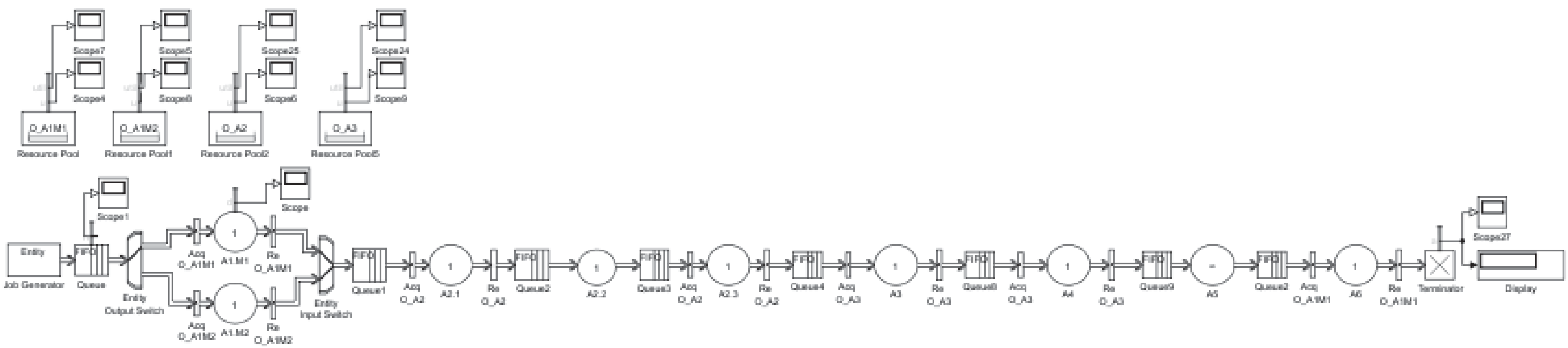


Figure A4.1: Simulation model of assembly line with operators

## A4.2 Model features

This appendix gives an insight into the features build in the simulation model and provides additional information for Section 6.4. Figure A4.2 shows the part of the simulation model which regulates the amount of workstations in process A1, namely routing message A1M2(cmd). This routing message is configured using the script shown in Figure A4.3. This figure also shows how a new day is modeled using the routing message JobDay(cmd). Another feature uses the routing message JobDay(cmd), namely the desired amount of products produced per day.

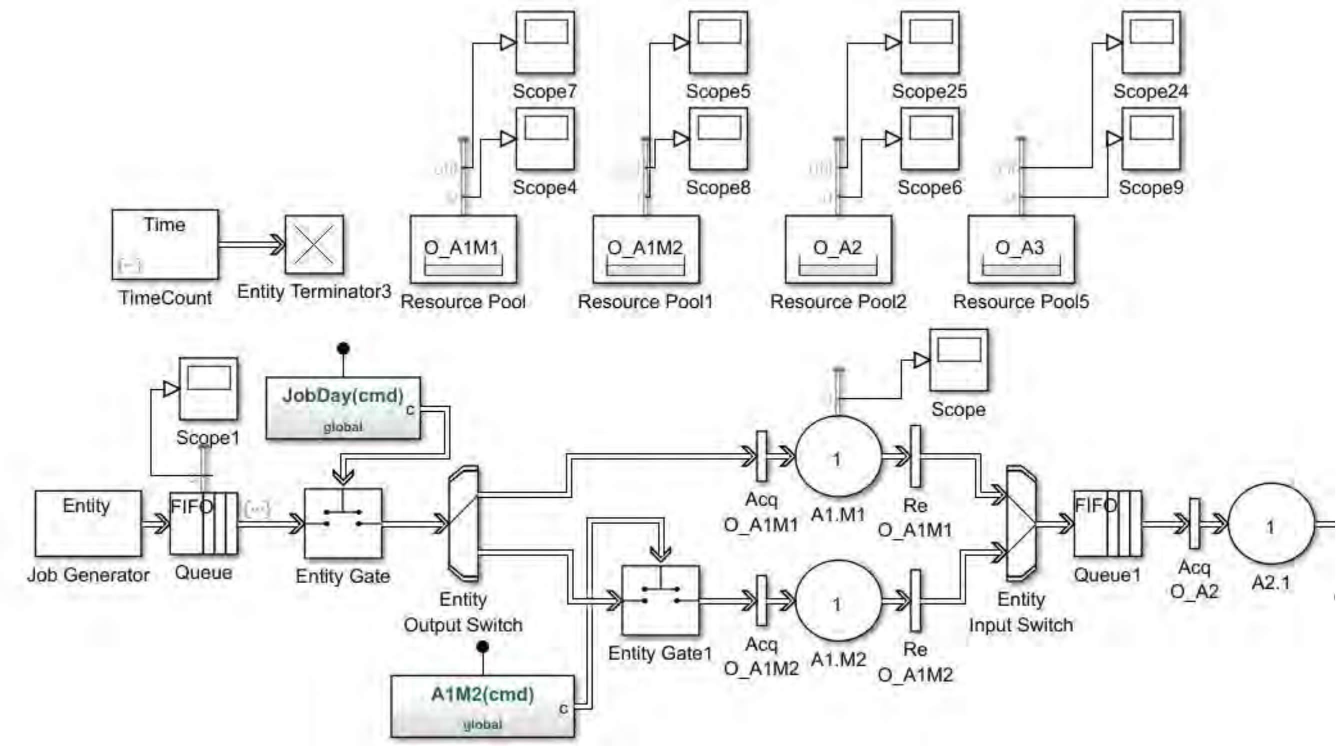


Figure A4.2: Expansion of simulation model with features

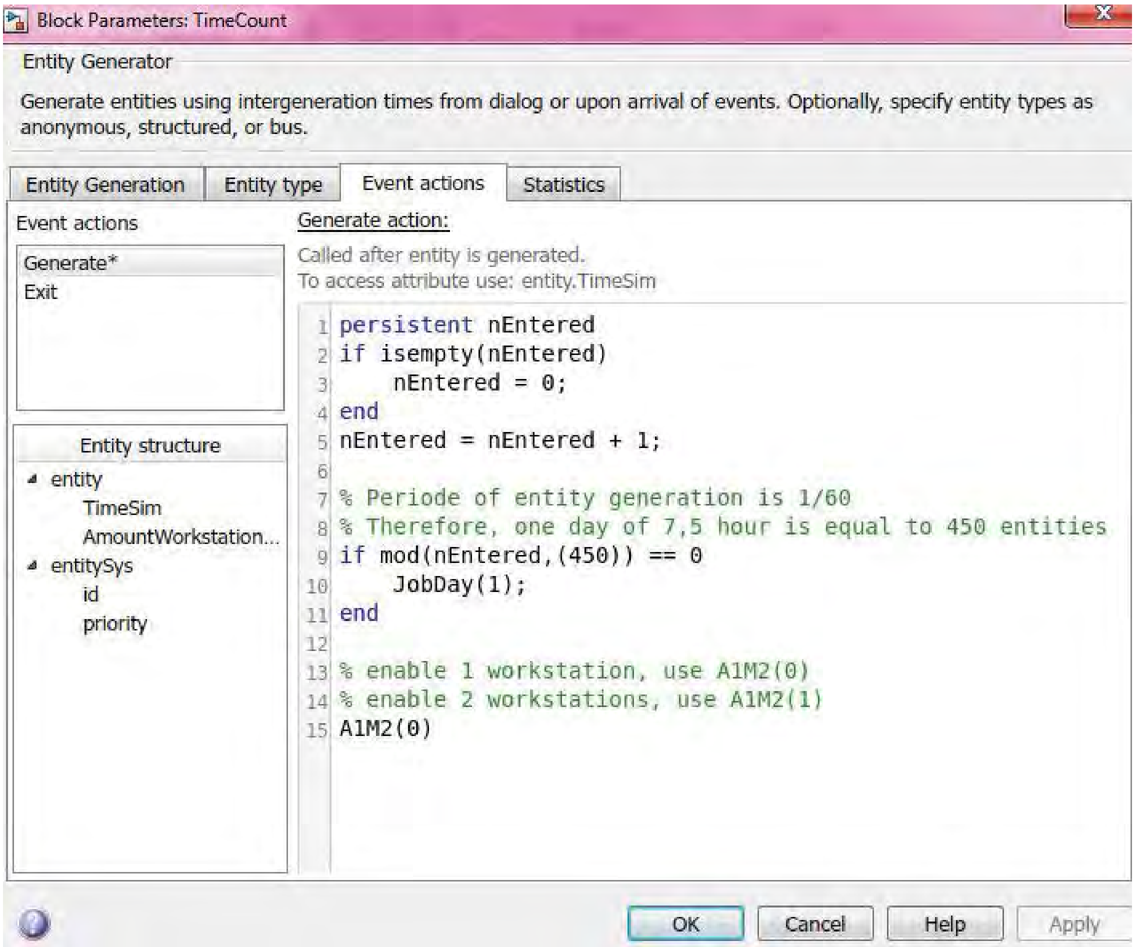


Figure A4.3: Parameters of the TimeCount block

## A4.3 Complete model

This appendix shows the complete Simevents model divided into three parts. This appendix provides additional information for Section 6.5. Figure A4.4 and Figure A4.5 must be placed consecutively and Figure A4.6 is placed below.

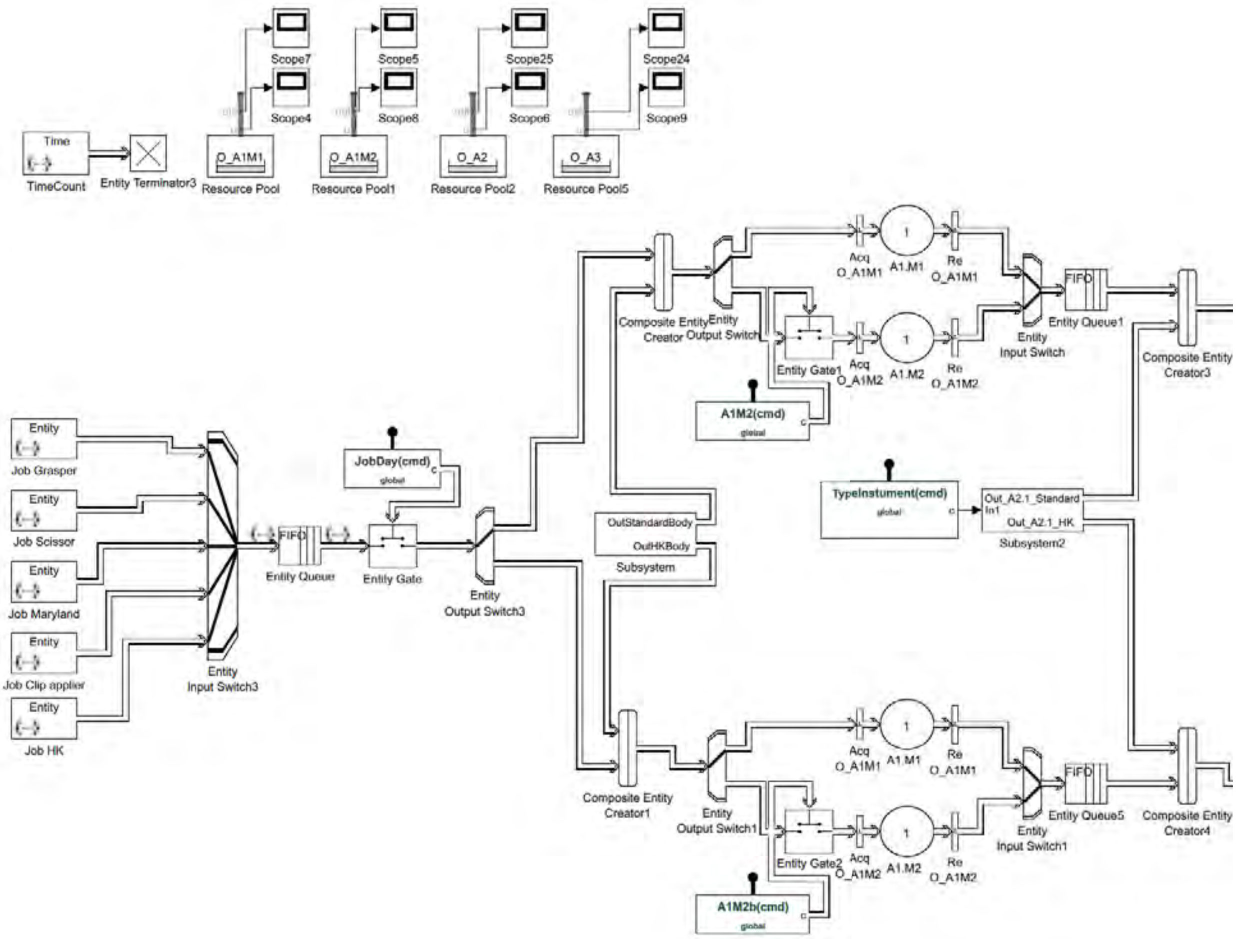


Figure A4.4: Complete simulation model of assembly line with operators and components

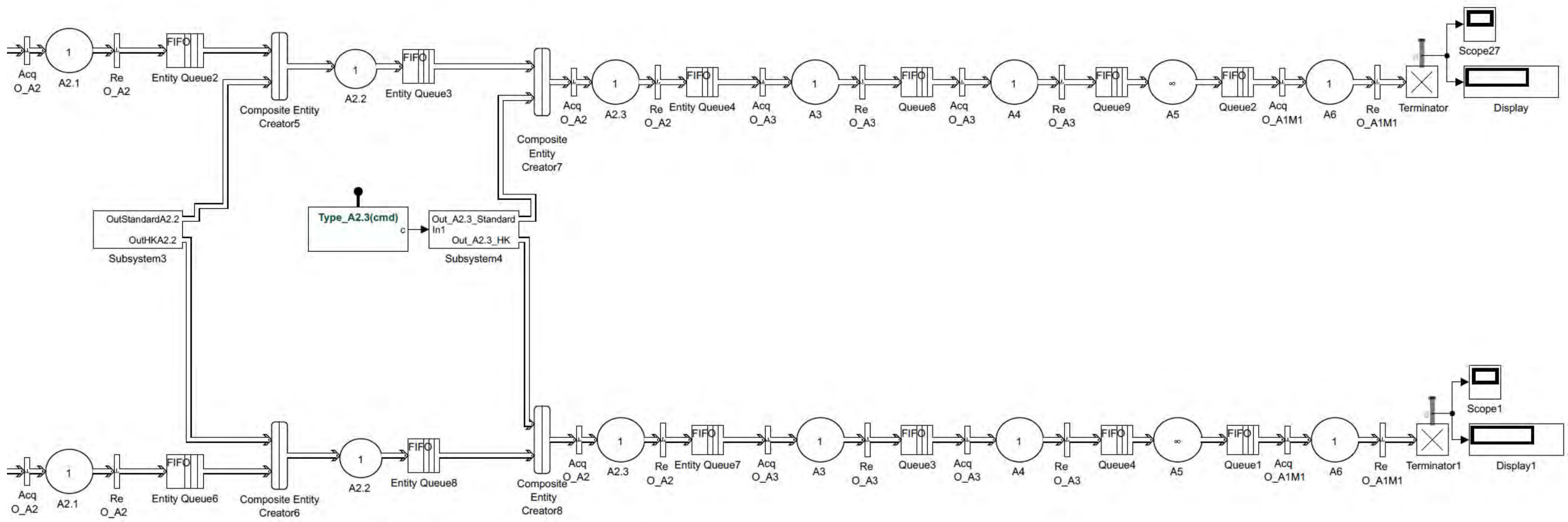


Figure A4.5: Complete simulation model of assembly line with operators and components

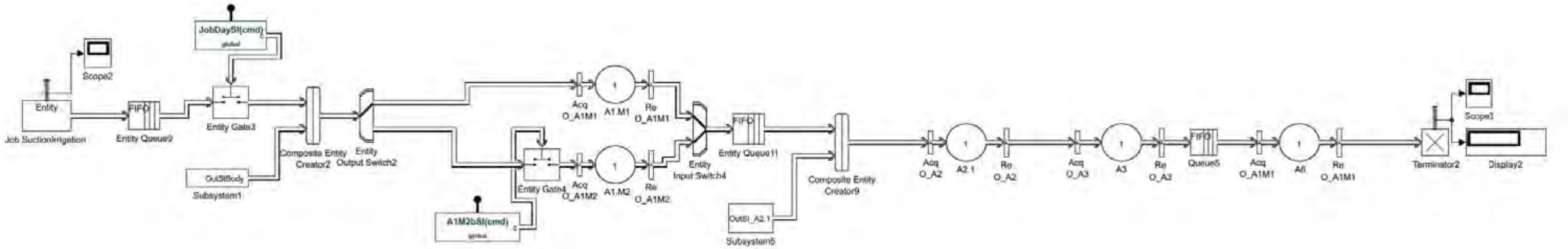


Figure A4.6: Complete simulation model of assembly line with operators and components

## A4.4 Pre-loaded entity generator

This appendix shows the matlab script to create a pre-loaded entity generator and provides additional information for Section 6.5. Figure A4.7 shows the script which creates an array of 4999 entities at time 0. Figure A4.8 shows the graph of created entities over time. As can be seen, the matlab script results in a pre-loaded entity generator. This pre-loaded entity generator can be used to model sufficient components at simulation start.

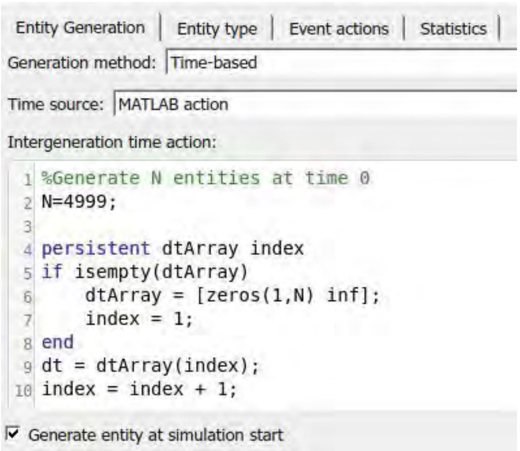


Figure A4.7: Matlab script for a pre-loaded entity generator

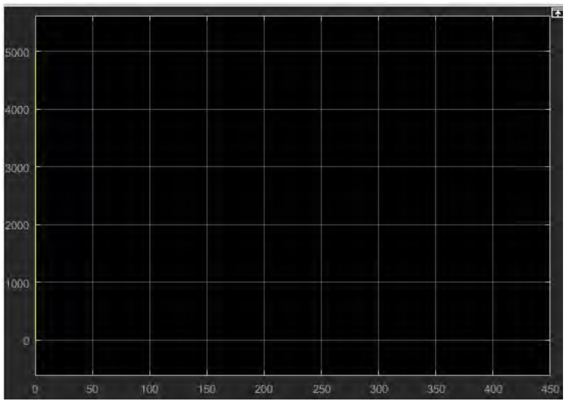


Figure A4.8: Scope of the amount of created entities over time

## A4.5 Subsystem of complete model

This appendix shows subsystem 4 and provides additional information for Section 6.5. Subsystems are created in the complete model, shown in Appendix A4.3, for the supply of components. Initially, the component generators are programmed infinite. Part of this subsystems is directed using routing messages. The status of the routing message is generated in the attributes of the jobs. This subsystem has two outputs, namely the output for the set of components for a standard body or a hook-knife body. As can be seen, an entity input switch uses the input 'In1' to choose the corresponding component using the attribute of the entity. Leading to an assembly of the right set of components.

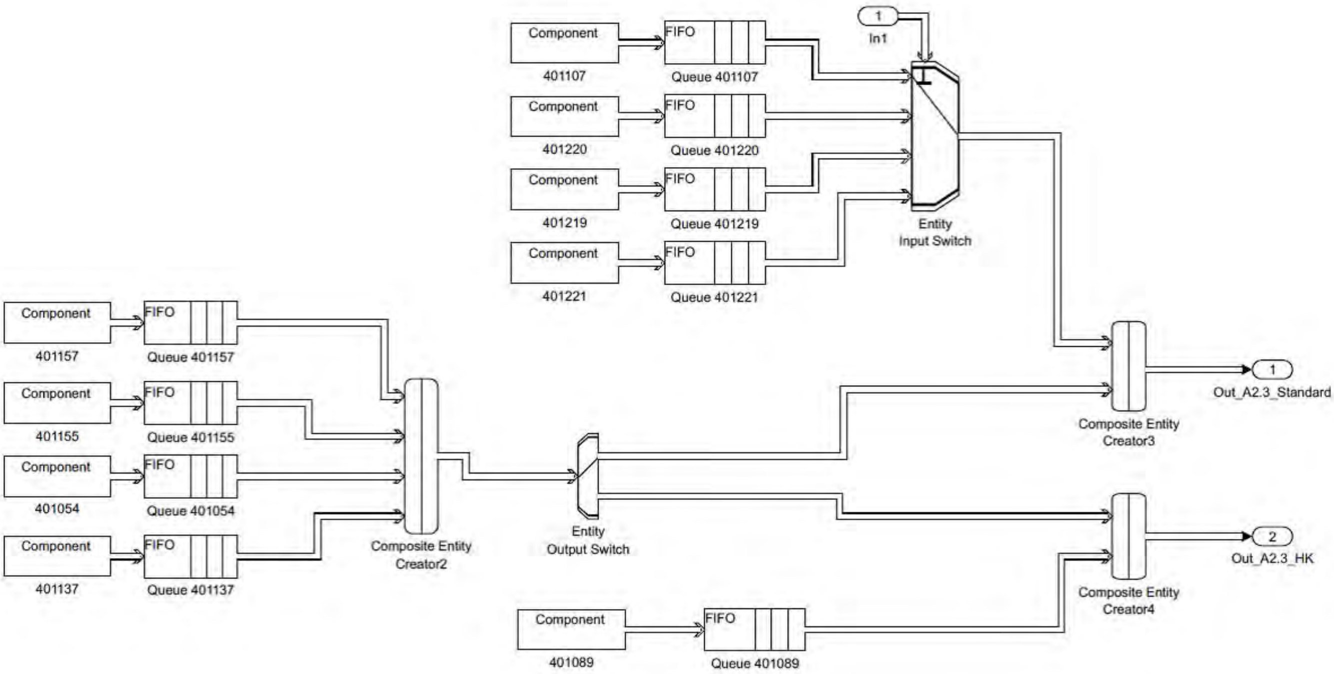


Figure A4.9: Subsystem 4 of the complete model

## A4.6 Partial validation model

This appendix shows the partial model of process A1 with the $(Q, s)$ policy implementation and provides additional information for Section 6.5.1. As can be seen in Figure A4.10, only process A1 is modeled. If the amount of components in the queue is smaller or equal to 171, a message is sent to the entity gate and one bin of 637 components is released. Display 1 shows the average queueing length of 388.2 products and the other display shows the amount of bins ordered during simulation time of 10000 hours.

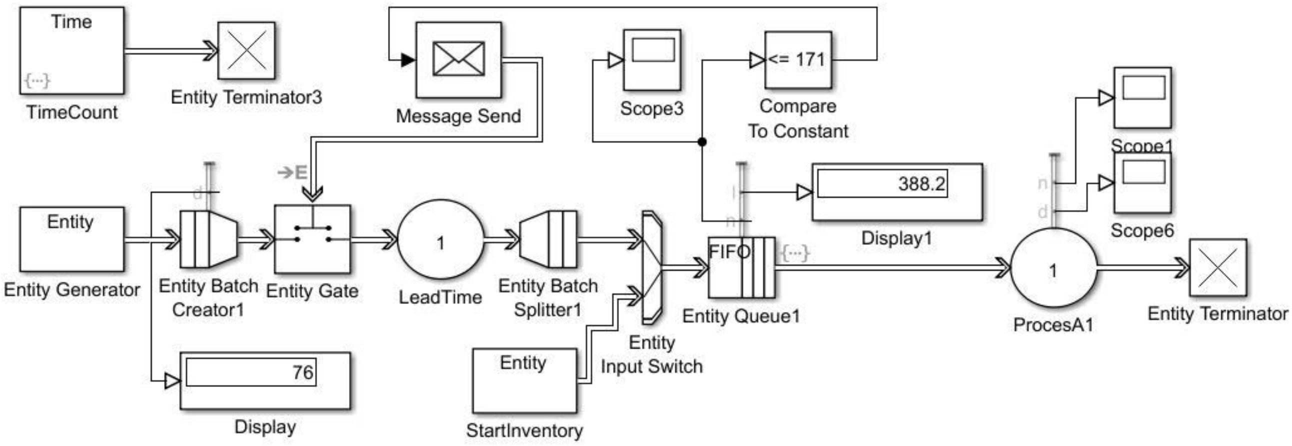


Figure A4.10: Partial model with process A1 with the $(Q, s)$ policy implementation.