Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria dell'informazione

Application of Model Predictive Control in Supply Chain Processes

Ph.D. Dissertation of:
ing. Chalakkal Varghese Kishore

Advisor:
Prof. Conte Giuseppe

Coadvisor:
Prof.ssa Perdon Anna Maria

XXXII Cycle

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria dell'informazione

Application of Model Predictive Control in Supply Chain Processes

Ph.D. Dissertation of:
ing. Chalakkal Varghese Kishore

Advisor:
Prof. Conte Giuseppe

Coadvisor:
Prof.ssa Perdon Anna Maria

XXXII Cycle

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

…a Sandra, Stefano e Bibi

Acknowledgments

I would like to thank Prof. Giuseppe Conte my advisor and above all Prof.ssa Anna Maria Perdon whose guidance was pivotal to this research, with her precious comments, help and encouragement.

I would like to thank my wife Sandra and my whole family for putting up with me and my professional and academic commitments. Only they know how much work has gone into this.

I would like to thank two special teachers of my school years, smt. Shobha Menon and smt. Usha Kumari P, for forging in me the scientific method and teaching me to think out of the box. *Gurave Namaha.*

Finally I would like to also like to thank each and every one I met in my career in supply chain, who taught me the importance of dirty work and led me to many questions that resulted in this research work. I would also thank all the companies where I worked.

Ancona, March 2020

ing. Chalakkal Varghese Kishore

Abstract

This thesis views the Material Requirements Planning (MRP) from an uncommon perspective of matrices. The whole process is developed using a set of matrices evolving over time in a time variant system approach. Instead of iterating along Bill Of Material (BOM) levels we will simultaneously calculate the materials requirement for all products at any given instance of time.

The main advantage of this approach is the speed: we can calculate MPS and MRP in seconds.

In the development of this idea we will be following a model predictive control approach, moving along the framework of SIOP (Sales, Inventory and Operations Planning, starting with a detailed analysis of demand planning concepts and techniques. We will then develop in detail the core concepts of the matrix approach to material requirements calculation, starting with Master Production Schedule (MPS).

We will extend this approach to the next step, the Material Requirements Planning (MRP) where we will see how the demands for the single items are further *exploded* down to the components of that item. In a multi-product industry with complex products and with components that could be part of more than one product, this calculation though conceptually simple become a heavily complex job. Change in product structure, change in bill of materials to say it in a more technical term, obsolescence and new product introductions further complicates this calculation. Instead of the iterative approach widely used in literature and all current software applications, we will use a matrix approach here also. Instead of calculating the requirements item per item and then summing it up, the proposed matrix structure will do the calculations for all the items for a specific time period all at once.

With the Master Production Schedule and Material Requirements Planning calculated, we will also extend this matrix approach to calculate the inventory levels and capacity requirements. While calculating inventory levels we will also see an important and direct application of this method in calculating the stockouts. In calculation of capacity requirement we will focus specifically on how direct labour requirements are calculated using the matrices.

A last and important application of system modelling is in financial planning, especially on a systems approach to stockout forecasting.

Contents

1. Introduction **1**

1.1. Model Predictive Control 2

1.1.1. MPC Strategy 3

1.1.2. Advantages and Disadvantages 5

1.2. Supply Chain 5

1.2.1. Supply Chain Management 6

1.2.2. History and Developments 6

1.3. Sales, Inventory and Operations Planning 7

1.3.1. SIOP vs S&OP 8

1.4. Relevance of MPC in supply chain processes 9

2. Demand Planning **11**

2.1. Time Series and Demand Forecasting 11

2.2. Qualitative Demand Forecasting Techniques 13

2.2.1. Survey of Customers 13

2.2.2. Sales Force Opinion 13

2.2.3. Delphi Method 14

2.3. Quantitative Demand Forecasting Techniques 15

2.3.1. Time Series 15

2.3.2. Error in Forecasting 15

2.3.3. Naive Method 16

2.3.4. Linear Regression 17

2.3.5. Average Method 17

2.3.6. Exponential Smoothing 17

2.3.7. Level and Trend 19

2.3.8. Double Exponential Smoothing 21

2.3.9. Triple Exponential Smoothing 23

2.4. Conclusion 25

3. Inventory Planning: Master Production Schedule **27**

3.1. Master Production Schedule 28

3.1.1. Time Fence 29

3.1.2. Planning Horizon 30

3.2. MPS Table 30

3.3. Master Production Schedule Calculation 31

3.4. Matrix Approach to Master Production Schedule 35
3.5. MPS Matrix for Single Item 36
3.6. Multidimensional MPS Matrix 39
3.7. Multidimensional MPS Matrix Calculation: Example 43
3.8. MPS Matrix Calculation: Results 45
3.9. Conclusion 49

4. Inventory Planning: Material Requirements Planning 53
4.1. Inputs for Material Requirements Planning 53
4.2. Steps in Material Requirements Planning 54
4.2.1. Calculating Gross Requirements 55
4.2.2. Calculating Net Requirements 55
4.2.3. Considering Lead Times 56
4.2.4. Calculating a list of requirement for single components . 56
4.2.5. MRP Table 56
4.3. Material Requirements Planning 58
4.3.1. Material Requirements Planning: Process 58
4.3.2. Material Requirements Planning: Example 61
4.4. Multidimensional Matrix Approach to Material Requirements Planning 65
4.4.1. Requirements Calculation using Matrices 65
4.4.2. Expanded Requirements Calculation using Matrices . . 69
4.4.3. Expanded Requirements Calculation using Multidimensional Matrices 75
4.4.4. Test results 82
4.5. Advantages 86
4.6. Conclusion 87

5. Inventory Planning: Inventory Management 89
5.1. Modelling Inventory using Matrices 90
5.2. Application of Matrix Approach 92
5.3. Use 97
5.4. Implementation 100
5.5. Conclusion 102

6. Operations Planning 103
6.1. Capacity Requirements Planning 103
6.1.1. Basic Terms 103
6.1.2. Direct Labour Capacity Table 104
6.1.3. Matrix Representation of DL Capacity Table 107
6.2. Multidimensional DL CRP Matrix 112
6.3. Conclusion 115

7. MPC in Financial Planning **117**
7.1. Income Statement . . . 118
7.2. Balance sheet . . . 118
7.3. Statement of Cash Flows . . . 119
7.4. Modelling the Cashflow Forecasting . . . 120
7.4.1. Signal Theory Approach . . . 121
7.5. Cash Flow Forecasting Using Traditional Methods . . . 122
7.6. Cash flow forecasting using distribution fitting method . . . 123
7.7. System Identification . . . 128
7.8. Comparison with Real Data . . . 129
7.9. Results . . . 131
7.10. Conclusion . . . 132

8. Conclusion **135**

A. Concepts and Definitions **139**
A.1. Random variables and Probability distributions . . . 139
A.2. Mean, Variance and Standard Deviation . . . 140
A.3. Normal Distribution . . . 140
A.4. Lognormal Distribution . . . 145
A.5. Logistics Distribution . . . 146
A.6. Probability Distribution fitting . . . 148
A.7. Pareto Chart . . . 148
A.8. KOSU . . . 148
A.9. Glossary of Financial Terms . . . 149

B. MCodes **151**
B.1. MPS Calculation using Multidimensional Matrices . . . 151
B.2. MRP Calculation using Multidimensional Matrices . . . 152
B.3. Capacity Planning Multidimensional Matrices . . . 156
B.4. Inventory Planning using Multidimensional Matrices . . . 158

Chapter 1.

Introduction

A supply chain can be seen as a network of facilities that procures raw materials, occupies with its transformation into intermediate and end products, their distribution and selling to the end customers [72]. On the other hand we have a well developed area of modelling and system theory applied mostly to other engineering areas. This research attempts a rendezvous of these two areas, to apply those the powerful concepts of modeling and simulation of dynamic systems theory to supply chain. This is done from a practical, pragmatic point of view derived from almost two decades of daily work in supply chain and logistic processes in heavily automated manufacturing facilities.

The presentation of this approach will be moving along the framework of SIOP (Sales, Inventory and Operations Planning - section 1.3 starting with a detailed analysis of demand planning in chapter 2 where we will discuss the main concepts and techniques used in demand planning. We will first analyze qualitative methods and then pass on to quantitative methods.

In the development of this idea we will be following a model predictive control approach, where we create a process model and then apply all known input and output values to this model for the entire prediction horizon. Of all the future values of the output calculated, we will be using only a recent set of values and the process is repeated at the next time instance. A more detailed insight could be seen in section 1.1.

In the next three chapters the core concepts of the matrix approach to material requirements calculation are discussed in detail. In chapter 3 we will see in detail the Master Production Schedule (MPS). It contains a statement of the volume and timing of the end products to be shipped to the customers. When there are many items, many customers and a large time horizon, the calculation quickly becomes complex. From the usual iterative approach for single items we will represent the whole set of items using a matrix and then do all calculations on this matrix.

We will extend this approach to the next step, the Material Requirements Planning (MRP) in chapter 4. Here we will see how the demands for the single item are further *exploded* down to the components of that item. In a multi-product industry with complex products and with components that could be part of more than one product, this calculation though conceptually simple become a heavily complex job. Change in product structure, change in bill of materials to say it in a more technical term, obsolescence and new product introductions further complicates this calculation. Instead of the iterative approach widely used in literature and all current software applications, we will use a matrix approach here also. Instead of calculating the requirements item per item and then summing it up, the proposed matrix structure will do the calculations for all the items for a specific time period all at once.

With the Master Production Schedule and Material Requirements Planning calculated, we will also extend this matrix approach to calculate the inventory levels 5 and capacity requirements in chapter 6. While calculating inventory levels we will also see an important and direct application of this method in calculating the stockouts. Stockout refers to the inability to satisfy the customer demand with the current stock. In calculation of capacity requirement we will focus specifically on how direct labour requirements are calculated using the matrices. By direct labour we mean all the labour costs involved directly in the conversion of raw materials to finished goods.

A last and important application of system modelling is in financial planning, especially on a systems approach to cash flow forecasting, as detailed in chapter 7.

1.1. Model Predictive Control

Model predictive control (MPC) refers to a range of control methods which make explicit use of a model of the process to obtain the control signal by minimizing an objective function. It does not refer explicitly to any specific control strategy. Model predictive controllers rely on dynamic models of the process, most often linear empirical models obtained by system identification. This strategy started to appear in literature since late 70s [4].

The main idea behind is to optimize the current timeslot keeping future timeslots in account. This is achieved by optimizing a finite time-horizon, but only implementing the current timeslot and then optimizing again, repeatedly, thus differing from Linear-Quadratic Regulator (LQR). Also MPC has the ability to anticipate future events and can take control actions accordingly. PID controllers do not have this predictive ability [54][55].

MPC contains the following core ideas [20]:

- Explicit use of a model to predict the future time horizon output of the process
- Calculation of a control sequence to optimize a performance index

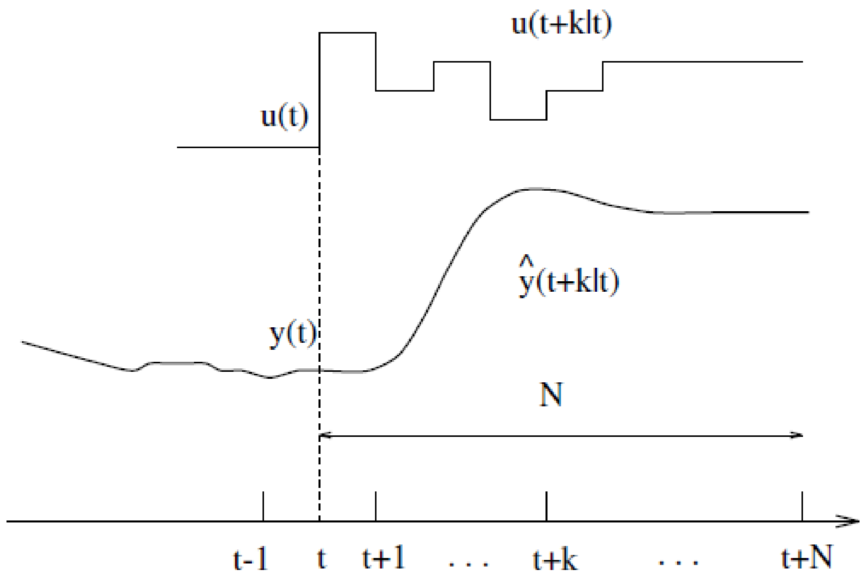

Figure 1.1.: MPC Strategy

- A receding horizon strategy, such that at each instant the horizon is moved towards the future applying only the first control signal of the sequence calculated in each step. Calculations are repeated starting from the new current state[54].

1.1.1. MPC Strategy

We can use the following strategy to characterize all controllers belonging to MPC family [20], represented in figure 1.1

1. The predicted future outputs $\hat{y}(t+k|t), k = 1, ..., N$ for the prediction horizon N are calculated at each instant t using the process model. This depends on known values of inputs and outputs up to the instance t including the current output $y(t)$, which is the initial condition, and on future control signals $u(t+k|t), k = 0, ..., N-1$ to be calculated. By $u(t+k|t)$ we mean the value of u at time instant $t+k$ calculated at time instant t.

2. The sequence of future control signals is computed to optimise a performance criterion to minimize the error between a reference trajectory and the predicted process output [93]. Normally the control effort is included in the performance criterion.

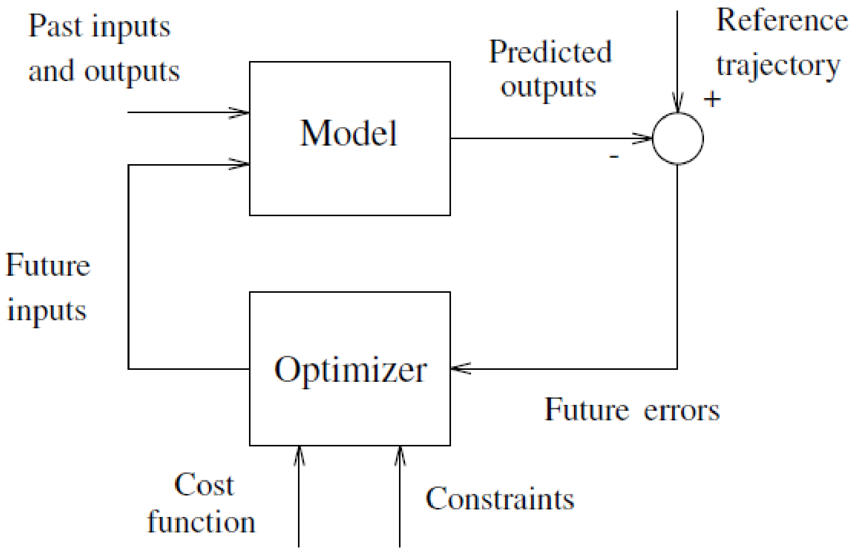

Figure 1.2.: Basic structure of MPC Process

3. Only the current control signal $u(t|t)$ is transmitted to the process. At the next sampling instance $y(t+1)$ the measurement is done once again and step 1 is repeated and all sequences brought up to date. Thus $u(t+1|t+1)$ is then calculated using the receding horizon concept.

In order to implement this strategy, the basic structure shown in Figure 1.2 is used. A model is used to predict the future plant outputs, based on past and current values and on the proposed optimal future control actions. These actions are calculated by the optimizer taking into account the cost function (where the future tracking error is considered) as well as the constraints [56] [83].

Camacho and Bordons [20] use an analogy of driving a car to illustrate the concept behind MPC strategy. The driver knows the desired reference trajectory for a finite control horizon and by taking into account the car characteristics (mental model of the car) decides which control actions (accelerator, brakes, steering) to take to follow the desired trajectory.

Only the first control actions are taken at each instant, and the procedure is repeated for the next control decision in a receding horizon fashion. In the classical control schemes, such as PIDs, the control actions are taken based on past errors. If the car driving analogy is extended, the PID way of driving a car would be equivalent to driving the car just using the mirror as shown in Figure 1.3

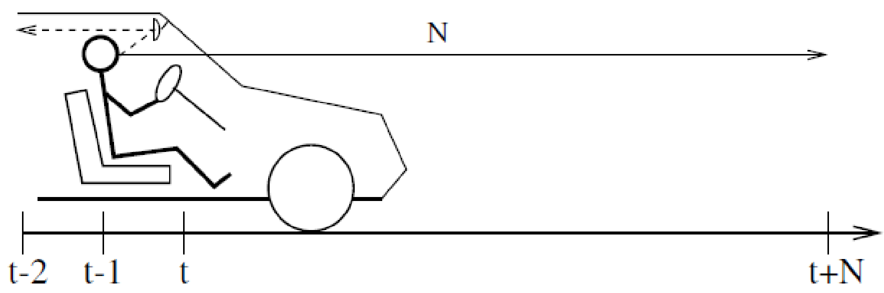

Figure 1.3.: MPC Car Driving Analogy

This analogy is not totally fair with PIDs, because more information (the reference trajectory) is used by MPC. Notice that if a future point in the desired reference trajectory is used as the setpoint for the PID, the differences between both control strategies would not seem so abysmal.

1.1.2. Advantages and Disadvantages

Model predictive control offers several important advantages [2]:

- Intuitive and attractive concepts for industry applications
- Useful in controlling a wide variety of processes including those with non minimum phase, long time delay or open loop unstable characteristics
- Deal with single-input single-output,multi-variable, multi-input multi-output processes
- Easily applicable to batch processes like demand forecasting where the future reference signals are known
- accurate model predictions can provide early warnings of potential problems

The significant disadvantage is that this strategy requires an accurate model of the process to be controlled.

1.2. Supply Chain

The Council of Supply Chain Management Professionals (CSCMP) define the supply chain [96] as the network

- starting with unprocessed raw materials and ending with the final customer using the finished goods, the supply chain links many companies together;

- the material and informational interchanges in the logistical process from the acquisition of raw materials to the delivery of finished products to the end user. All vendors, service providers and customers are links in the supply chain [51].

1.2.1. Supply Chain Management

Supply Chain Management encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities [51] [89]. Supply chain management integrates supply and demand management within and across companies. It includes all of the logistics management activities, manufacturing operations, and it drives coordination of processes and activities with and across marketing, sales, product design, finance, and information technology. [96]

1.2.2. History and Developments

Over the last century, supply chain management has evolved from an initial focus on improving relatively simple, but very labor-intensive processes to the present day engineering and managing of extraordinarily complex global networks. We can identify the following milestones in the development of supply chain: [81]

- Fredrick Taylor, considered the father of industrial engineering, focused his early research on how to improve manual loading processes. [91]

- In the 1940s and 1950s, the focus of logistics research was on how to use mechanization (e.g., pallets and pallet lifts) to improve the very labor intensive processes of material handling and how to take better advantage of space using racking and better warehouse design and layout.

- In the mid 1950s, this concept was extended to transportation management with the development of inter-modal containers together with ships, trains, and trucks to handle these containers.

- By the 1960s, a clear trend had developed in shifting more time-dependent freight transportation to truck rather than rail. This led to the need for joint consideration of warehousing, material handling, and freight transportation, which emerged under the label of "Physical Distribution" [21].

- The 1980s marked the beginning of a sea-change in logistics in the history of supply chain management. The emergence of personal computers in the early 1980s provided tremendously better computer access to planners and a new graphical environment for planning [76] [35]. This spawned a flood of new technology including flexible spreadsheets and map-based interfaces which enabled huge improvements in logistics planning and execution technology.

- The logistics boom was fueled further in the 1990s by the emergence of Enterprise Resource Planning (ERP) systems. These systems were motivated in part by the successes achieved by Material Requirements Planning systems developed in the 1970s and 1980s, in part by the desire to integrate the multiple databases that existed in almost all companies and seldom talked to each other, and in part by concerns that existing systems might have catastrophic failures as a result of not being able to handle the year 2000 date [36] [92].

- Currently the communication capabilities have fundamentally changed the way we think about communications and information sharing. However, supply chain and logistics planning is still primarily based on the distributed models that came as the result of personal computers. There is no question that academic research can enable a new generation of supply chain and logistics planning technology

1.3. Sales, Inventory and Operations Planning

Sales, Inventory and Operations Planning (SIOP) is an integrated business management process which allows organizations to develop a supply and demand plan balanced to reduce out-of stock risks. In order to implement an efficient sales and operations planning process, an organization needs to bring together different elements within the business including sales, marketing, finance and operations. SIOP is a key business process that derives from an organization's strategic plan providing

- updated sales and new product development plans

- direct and indirect labour requirement

- production capacity capacity and allocation planning

- optimization of stock levels and replenishment

1.3.1. SIOP vs S&OP

S&OP and SIOP are two concepts that are often used interchangeably – which can be misleading. S&OP and SIOP are different levels of maturity in an organization – with SIOP being the more mature among the two. S&OP means Sales and Operations Planning.

S&OP processes will generally consist of the following meetings in a monthly cycle: Demand Planning - Supply Planning - Cross-Functional - Executive Review roughly as follows:

- Week 1 – Demand Planning
- Week 2 – Supply Planning
- Week 3 – Cross-Functional meetings and reviews
- Week 4 – Executive Review

APICS defines S&OP as the "The S&OP process develops tactical plans that assist management in strategically directing the business to achieve continuous competitive advantage. It integrates customer-focused marketing plans for new and existing products with the management of the supply chain. The process integrates all business plans into a single set that meets all needs of the functions of the business. The S&OP process is performed at least once a month and is reviewed by management at an aggregate level." [5]

Formal Sales Inventory and Operations Planning provides a single set of numbers and a routinized process to ensure that top management's objectives and plans are realistic and accurately reconciled to the detailed scheduling done in a company. The top executives and heads of all functional areas in the company must participate in this process, along with scheduling and marketing personnel [46].

The planning horizon for a typical S&OP process is long term typically from 18 to 36 months. The selection of a time horizon is an important decision and there are different factors that influence this decision including type of industry, product characteristics, and the time of the year when S&OP planning takes place. Additionally, the S&OP process is conducted at an aggregate level. The main focus is on the family of products and not on single items which together make up the family.

To have an effective S&OP process [43],

- The focus should be on how to change or better achieve future plans or forecasts, not on penalizing poor prior predictions.
- The focus should be on future months' plans (generally 3 or more months into the future). Short term planning is rigid and changes are often costly.

It is management's job to deal with the uncertainty of the future, and change forecasts and plans far enough in advance to better avoid short term emergencies.

- The meeting should deal only with overall rates of shipment and production. Individual issues should be discussed outside this meeting. Specifics should be discussed only if there is potential impact on meeting future plans.

- this process and the meeting should evolve to the point where middle management identifies problems and formulates suggested solutions before the meeting, so that top management's time can be preserved for only evaluating and approving the proposals.

A good Sales and Operations Planning process can start to produce benefits even before full MRP is operational [41]. It provides a single set of company numbers, maintained monthly and expressed at a summary level appropriate for top management review.

SIOP processes are built on top of this with a more financial influence.helping a better understanding of inventory levels and their financial impacts, cash flow etc. There tends to be a deeper sense of accountability, higher amounts of Executive exposure and support, and the process is overall viewed as a fundamental company backbone.

1.4. Relevance of MPC in supply chain processes

Model predictive control starts with model, thus the first important and relevant step is modeling the supply chain.

Master Production Schedule (MPS) contains a statement of the volume and timing of the end products to be made. A stable and well defined MPS translates into stable component schedules and improved performance in the plant. MPS can be considered as a statement of planned future output.

MRP systems help to determine what, how much, when to order as well as when to schedule delivery. They also help to plan and control inventory, plan capacity and priority for the production.

Chapter 2.

Demand Planning

Demand planning can be defined as the use of analytics — optimization, text mining, and collaborative workflow—to use market signals (channel sales, customer orders, customer shipments, or market indicators) to predict future demand patterns of the demand for an item or set of items[13].

Companies started giving importance to demand planning starting from 1980s and currently all Enterprise Resource Planning (ERP) softwares implement some form of demand planning. There are also many good softwares that focus on demand planning in a very specific manner. The forecasting period, the term used to refer to the period in which the demand is forecasted, can move normally from a 12 month horizon to a 18 month horizon. Demand planning gives best results at short to medium time periods. Qualitative methods are the best for forecasting longer time period. It is ultimately the company policy that decides the exact duration of the planning horizon.

In this chapter we will overview the main concepts and techniques used in demand planning. We will first analyze qualitative methods and then pass on to quantitative methods.

2.1. Time Series and Demand Forecasting

A *series* is any ordered sequence of values or data points which refer to some information. *Time series* is a series of these data points arranged in the order of their occurrence in time. The time references are usually equally spaced in time. When we say analysis of time series what we trying to do is to extract meaningful information from these data points. If we try to predict the future values for these data points then we are speaking of time series forecasting. There is a temporal order among these data points.

In a supply chain, *demand* refers to the request of one or more finished goods that a company may sell to its customers to generate profit. These need not always be material items. For example, a software company sells software that is not physical while a car manufacturer sells cars that are physical objects (with non physical objects also included like software).

For a physical product manufacturing company, we would need raw materials, machines and human labour to produce finished goods. This process is almost never instantaneous. So we can think of a difference in time between the time of the demand and the time of the actual production, which we will call *production lead time*. Lead time has many different connotations according to the point of view from which the entire supply chain is viewed. From the procurement point of view of the raw materials, we can speak of *purchasing lead time* or *procurement lead time*. This is the time difference between the time at which an order is placed to a supplier of raw materials or components to when the supplier actually delivers that order to the warehouse. We can speak of *production lead time* as the time required to produce a part once the production order is placed. *Supply lead time* on the other hand refers to the time between the order placed by a customer to when that item is physically supplied to that customer.

So any single order lives through all these lead times. When a customer places an order to the company, there is a time required to process this order to see if the company has everything needed to produce this order, which seldom is the case. So the company orders the missing components or raw materials to its suppliers, who in turn repeats this process to their respective suppliers(purchasing lead time) to supply the parts requested to the company. The company then produces the parts (production lead time) and then sends it to the customer, with a transportation lead time. All this means that to have an order in time, the whole network needs to respect the relative lead times, something almost close to impossible. For example, a car showroom does not have an exact knowledge of when a customer comes and purchases a car. Thus the showroom does not know exactly when to place an order to the car manufacturer, who in turn does not know exactly when to place orders for the parts and organize the production and so on. This brings us the need for forecasting the demands at all stages even without the actual order. We try to assume what could be the demand using various techniques which we will analyze in the following section.

Forecasting is attempting to predict future statistics, typically, demand or sales. It requires that all possible factors surrounding the decision-making process are recorded. Factors that affect forecasting include sales demand patterns, economic conditions, competitor actions, market research, product mixes, and pricing and promotional activities. Forecasts can be made at strategic, tactical, and operational levels[5].

Demand Forecasting is critical to all modern business since all strategic and operational plans are developed around it. All strategic and long-range plans of a business like budgeting, financial, sales and marketing plans, capacity planning, risk assessment and mitigation plans are all formulated starting from demand planning.

2.2. Qualitative Demand Forecasting Techniques

Qualitative forecasting techniques, as the name suggests, are qualitative in nature and are not based on any specific numerical algorithm or mathematical method. They rely heavily on expertise and thus are time-consuming and costly. On the plus side they are the best suited when we need to have long time predictions. These often are the only reliable methods available around.

We will now look at the main qualitative methods commonly used in demand forecasting. Of all the suggested methods in literature, we will focus on those commonly used in real world industries.

2.2.1. Survey of Customers

This is the basic, easiest and oldest of all. It consists of directly estimating from the customers what they would buy in the future. This could work for bulk products or small scale direct to customer businesses. The best estimate we could get is valid but useful only for a short term and the entire focus is on the customer and his sentiments without any specific numeric foundation.

2.2.2. Sales Force Opinion

Sales force opinion on market research is the next step from survey of customers. We go up a level and involve only the sales force assuming that they are the ones who have the best knowledge of their customers and their demands. so we start to speak of a collective aggregation. They are ideal for situations where little historic data is available or for the launch of new projects where the salesman's judgement is fundamental. Market research is aggregated with direct input from sales force. In case there is a disagreement among the sales force, a Delphi approach could be used to arrive at a consensus. We will discuss Delphi method in the next section.

The major drawback of this approach is the excessive reliance on subjective judgement and lack of factual or numerical basis. Another variation of this, called *Executive Judgement*, involves the executives from other areas also to augment the judgement of sales force. There is a third variant called *Express Opinion* where the opinion from experts of the field of activity are also augmented to that of sales force.

2.2.3. Delphi Method

The Delphi method was originally conceived in the 1950s by Olaf Helmer and Norman Dalkey of the Rand Corporation. The name refers to the Oracle of Delphi, a priestess at a temple of Apollo in ancient Greece known for her prophecies. The Delphi method allows experts to work towards a mutual agreement by conducting a circulating series of questionnaires and releasing related feedback to be further discussed in each subsequent iterations. The experts' responses shift as rounds are completed based on the information brought forth by other experts participating in the analysis [17]. The experts are allowed to adjust their answers in subsequent rounds, based on how they interpret the group response that has been provided to them. Since multiple rounds of questions are asked and the panel is told what the group thinks as a whole, the Delphi method seeks to reach the correct response through consensus[62]. Delphi is based on the principle that forecasts (or decisions) from a structured group of individuals are more accurate than those from unstructured groups [7].

Even though the technical and computational communities would consider qualitative approaches as naive, they could often be the only best in case of highly complex and dynamic markets like automotive sector, especially when Original Equipments (OE) demand lives along with Original Spares and Aftermarket. Quantitative approaches does not forecast well the demands in such conditions. In cases where they could, the computational effort and knowledge requirement outweighs the advantages.

Let us take a real world scenario of an automotive component manufacturer who wants to forecast their demand. When a car starts its life cycle there is a ramp up exclusively of new cars. If the model is a success then this ramp-up is followed by an immediate spike to then follow a growing trend. Towards the end of its life, the sales volume starts to drop exponentially to end with the end of life of the car. Along with this, there is also the demand for spares that follows an entirely different trend. First year(s) normally have low volume of about 2% of the total cars sold of that model till then which almost always comes from demand due to accidents. This percentage is normally inferred from the accident rate of that particular model and related markets often by Delphi method. This then grow around the second year to go to the peak around the end of the OE life cycle almost always following the trend of the accident rate of that particular model. Then there is a a gradual decrease, first starting from around 10% of the total number of cars sold at the end of life, to come down exponentially with the last cars nearing their technical end of life, ramping down about 5% less volume every year. In the final years, there will be a heavy replacement mostly due to scrapping. So the scrap rate is also important in spare part calculations. The market reception or the pulse of the market completely defines how demand progress during the successive years. A good sales team is really precious in this area more than any algorithm. In this specific example from an real world industry, the Delphi method was used to arrive at this conclusion.

2.3. Quantitative Demand Forecasting Techniques

Quantitative methods are used when reliable historic data is available with clear and stable relationships and trends. The basic assumption is that the past and present reflects into the future. This could be true for short or medium period. When there are significant changes in the trend, these techniques should be supplemented by other approaches. The quantitative approaches could range from naive approach of considering the next data point to be same as the current one to more elaborate forecasting techniques.

2.3.1. Time Series

Let us start with the concept of *series* mentioned earlier, which can be considered as a set (x, y) where x is the order. Since a *time series* is a series of data points indexed in the order of time, we will consider that x increments by 1 in our time series. In a time series, x will be the time and y the relative data point at that time point. Since forecasting tries to estimate the unknown values based on known values, we will refer to them as *observed* and *expected* values respectively. So the known, observed, data points will be $y_1, y_2, ...$ and expected values will be $\hat{y}_1, \hat{y}_2, ...$

2.3.2. Error in Forecasting

By error we mean the difference between observed and forecasted values. Following are the different measurements of errors normally used in supply chain:

Relative and Absolute Errors

Relative error is the simple difference between forecasted and actual values. Since this could be a positive or a negative value, we also use the concept of absolute error which is the absolute value of the difference, which is always positive.

Sum of Squared Errors (SSE) and Mean Squared Error (MSE)

When we consider the full series, the squared errors are typically summed up giving us the concept of Sum of Squared errors (SSE). Mean Squared Error

(MSE) can be calculated as:

$$\frac{1}{N}\sum_{k=1}^{N}(\mid F_k - A_k \mid)^2 \tag{2.1}$$

Mean Absolute Deviation (MAD)

Mean Absolute Deviation (MAD) can identify which high-value forecasts are causing higher error rates. MAD takes the absolute value of forecast errors and averages them over the whole forecast horizon. Taking an absolute value avoids the positives and negatives cancelling each other out. It is calculated as in equation 2.2:

$$\frac{1}{N}\sum_{k=1}^{N}\mid F_k - A_k \mid \tag{2.2}$$

Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) also known as Mean Absolute Percentage Deviation (MAPD) expresses the accuracy as a percentage. It is given by the equation 2.3

$$\frac{1}{N}\sum_{k=1}^{N}\left|\frac{F_k - A_k}{A_k}\right| \cdot 100 \tag{2.3}$$

It cannot be used if there are zero y values and for forecasts which are too low. The percentage error cannot exceed 100%, but for forecasts which are too high there is no upper limit to the percentage error. MAPE puts a heavier penalty on negative errors than on positive ones. This could be mitigated using another metric called Mean Arctangent Absolute Percentage Error (MAAPE) which was developed through looking at MAPE from a different angle. In essence, MAAPE is a slope as an angle, while MAPE is a slope as a ratio [88].

2.3.3. Naive Method

This is the simplest of all according to which the data point at the first point in the future is the same as that of the last known data point. this can be expressed as in equation 2.4:

$$\hat{y}_{n+1} = y_n \tag{2.4}$$

Even though it is simple, many companies still use this naive method with an additional markup. For example if the sales for this month was 100 cars, then the sales forecasted next month could be with a +10% markup, giving 110 cars.

2.3.4. Linear Regression

Linear regression is a linear approach of modelling the relationship between a dependent variable and one or more or independent variables. In case of a single variable this is called simple linear regression. Linear regression develops the equation of a line $y = a + b(x)$ that best fits a set of known data points (x, y) . Once the line is developed, the future x values can be plugged in to predict y, usually the demand.

For time series models, x is the time period for which we are forecasting. This method is ideal for picking up levels and trends in time series data. So We will be using it heavily in double and triple exponential smoothing methods.

2.3.5. Average Method

Instead of taking the last known data point as it is or with a $\pm$ markup, this approach takes an average of the last values or a subset of last data points as the estimate of the value at the next time point. This could be a simple average of all data points as in equation 2.5, where n is the total number of all known data points, or a moving average where we take a subset of data points with n members or a weighted average as in equation 2.6. In weighted average, w_i is the weight of the i^{th} element with recent points given more weight since they could be more relevant, the sum of total weights adds up to 1.

$$\hat{y}_{n+1} = \frac{1}{x} \sum_{i=1}^{x} y_i \tag{2.5}$$

$$\hat{y}_{n+1} = \frac{1}{x} \sum_{i=1}^{x} w_i y_i \tag{2.6}$$

Figures 2.1 and 2.2 gives an example of how this method works.

The averaging methods are not useful if there are trends or seasonality involved, where we should use the methods discussed in the next sections.

2.3.6. Exponential Smoothing

Exponential smoothing is a particular form of weighted average. Here we consider all the data points, progressively assigning exponentially smaller weights as we go back in time. In the simple moving average all the past observations are weighted equally. The weights adds up to e, decaying in an uniform manner. What we essentially do is a weighted average with two weights α and $1 - \alpha$ the

Data point	Demand	Level	Forecast	Error	Absolute Error	MSE	MAD	% Error	MAPE
0									
1	8.000								
2	13.000								
3	23.000								
4	34.000	19.500							
5	10.000	20.000	19.500	9500	9.500	90.250.000,00	9.500,00	95,00	95,00
6	18.000	21.250	20.000	2000	2.000	47.125.000,00	5.750,00	11,11	53,06
7	23.000	21.250	21.250	-1750	1.750	32.437.500,00	4.416,67	7,61	37,91
8	38.000	22.250	21.250	-16750	16.750	94.468.750,00	7.500,00	44,08	39,45
9	12.000	22.750	22.250	10250	10.250	96.587.500,00	8.050,00	85,42	48,64
10	13.000	21.500	22.750	9750	9.750	96.333.333,33	8.333,33	75,00	53,04
11	32.000	23.750	21.500	-10500	10.500	98.321.428,57	8.642,86	32,81	50,15
12	41.000	24.500	23.750	-17250	17.250	123.226.562,50	9.718,75	42,07	49,14

Figure 2.1.: Moving Average

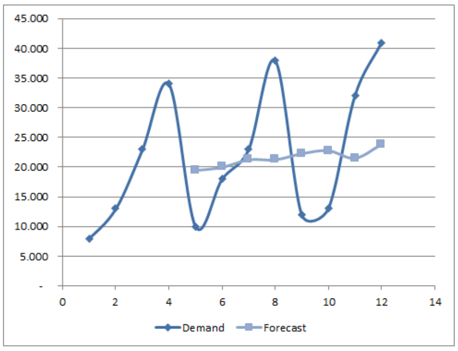

Figure 2.2.: Moving Average Graphical Representation

sum of which adds up to 1. The forecasted value $\hat{y}$ is given by the equation 2.7. α is called the *smoothing factor*. Exponential smoothing was first suggested in the statistical literature without citation to previous work by Robert Goodell Brown in 1956 [80], and then expanded by Charles C. Holt in 1957 [37].

$$\hat{y}_{x+1} = \alpha y_x + (1 - \alpha)\hat{y}_{x-1} \tag{2.7}$$

This formulation is attributed to Brown and is known as *Brown's simple exponential smoothing*.

$(1 - \alpha)$ is multiplied by the previous forecasted value and this goes on recursively till the beginning of the series or infinitely if there is none, giving the name *exponential* to this method. The value of α determines how much weight we give to the most recently observed values. Higher value of α means higher value to the left part of the equation. Each series has its own best α and the process of finding the best alpha is called *fitting*. Values of α close to 1 have less of a smoothing effect and give greater weight to recent changes in the data, while values of α closer to 0 have a greater smoothing effect and are less responsive to recent changes. There is no formally correct procedure for choosing α. Sometimes the statistician's judgment is used to choose an appropriate factor. We can also use the Method of Least Squares.

The exponential smoothing becomes useless in presence of other factors that influence the demand like promotions or price changes or competitive actions not regularly scheduled. Exponential smoothing puts substantial weight on past data points. So the initial value of demand will have an unreasonably large effect on early forecasts. This could be overcome by allowing the process to evolve for a reasonable number of periods (10 or more) and using the average of the demand during those periods as the initial forecast. We should note that the smaller the value of α, the more sensitive the forecast will be on the selection of this initial smoother value[58].

Figures 2.3 and 2.4 continues with the earlier example applying the exponential smoothing method applied to the same data now with $\alpha = 0,1$. Note that the first level is calculated from the average of known values and the successive using the formula 2.7.

2.3.7. Level and Trend

Till now we forecasted only a single data point into the future. To forecast more future data points we need to introduce a couple of new concepts: *Level* and *Trend*. Any give data points can be represented using a level and trend. Level is the point we calculated with earlier techniques , which is the y intercept. Level is often denoted by l. Trend the slope, which we algebraically define as $m = \frac{\Delta y}{\Delta x}$ with Δx and Δy the difference in x and y coordinates between two

					α	0,1			
Data point	Demand	Level	Forecast	Error	Absolute Error	MSE	MAD	% Error	MAPE
0		22.083							
1	8.000	20.675	22.083	14.083	14.083	198.340.277,78	14.083	176,04	176,04
2	13.000	19.908	20.675	7.675	7.675	128.622.951,39	10.879	59,04	117,54
3	23.000	20.217	19.908	- 3.093	3.093	88.936.486,34	8.284	13,45	82,84
4	34.000	21.595	20.217	- 13.783	13.783	114.196.859,90	9.659	40,54	72,27
5	10.000	20.436	21.595	11.595	11.595	118.246.640,77	10.046	115,95	81,00
6	18.000	20.192	20.436	2.436	2.436	99.527.532,15	8.777	13,53	69,76
7	23.000	20.473	20.192	- 2.808	2.808	86.435.713,79	7.925	12,21	61,54
8	38.000	22.226	20.473	- 17.527	17.527	114.031.549,65	9.125	46,12	59,61
9	12.000	21.203	22.226	10.226	10.226	112.979.314,95	9.247	85,21	62,45
10	13.000	20.383	21.203	8.203	8.203	108.410.264,71	9.143	63,10	62,52
11	32.000	21.544	20.383	- 11.617	11.617	110.824.073,79	9.368	36,30	60,14
12	41.000	23.490	21.544	- 19.456	19.456	133.132.064,78	10.208	47,45	59,08

Figure 2.3.: Exponential Smoothing

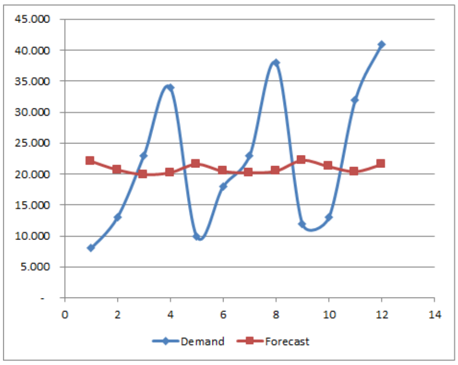

Figure 2.4.: Exponential Smoothing Graphical Representation

points. In a time series, Δx is always equal to 1. so we can define the trend b as in equation 2.8. To forecast trends we can use the exact same methods that we used for level

$$b = y_x - y_{x-1} \tag{2.8}$$

2.3.8. Double Exponential Smoothing

Double exponential smoothing is the application of exponential smoothing to both level and trend, the recursive application of an exponential filter twice [22]. The basic idea behind double exponential smoothing is to introduce a term to take into account the possibility of a series exhibiting some form of trend. This slope component is itself updated by means of exponential smoothing. So we will have a set of three equations. one for level, one for trend and one for forecast which combine the other two as in the following equations 2.9, 2.10 and 2.11 where β is the trend.

$$\ell_x = ay_x + (1-\alpha)(\ell_{x-1} + b_{x-1}) \tag{2.9}$$

$$b_x = \beta(\ell_x - \ell_{x-1}) + (1-\beta)b_{x-1}) \tag{2.10}$$

$$\hat{y}_{x+1} = \ell_x + b_x \tag{2.11}$$

Let us continue with the example above this time adding the trend factor also.

To calculate level and trend we can use the linear regression. For all x and the corresponding y we first calculate their averages μ_x and μ_y. Then we subtract them from respective x and y as shown in figure 2.5 to get p and q, from which we calculate $p \cdot q$ as well as p^2. From these values we can calculate level and trend as in equations 2.12 and 2.13, which gives the level and trend for starting point.

$$level = \frac{(x-\mu_x)\cdot(y-\mu_y)}{(x-\mu_x)^2} \tag{2.12}$$

$$trend = \mu_y - level \cdot \mu_x \tag{2.13}$$

Continuing with our earlier example, by applying the double exponential method, we get the results as in figure 2.6 with $\alpha = 0.1$ and $\beta = 0.2$. The graphical form is as in figure 2.4.

We need at least two points to calculate the initial trend and we can forecast two data points into the future using this method. To forecast more than two

	Data Point	Demand				
	x	y	$p=x-\mu_x$	$q=y-\mu_y$	p*q	p^2
	1	8.000	-5,5	- 14.083	77.458	30,25
	2	13.000	-4,5	- 9.083	40.875	20,25
	3	23.000	-3,5	917	- 3.208	12,25
	4	34.000	-2,5	11.917	- 29.792	6,25
	5	10.000	-1,5	- 12.083	18.125	2,25
	6	18.000	-0,5	- 4.083	2.042	0,25
	7	23.000	0,5	917	458	0,25
	8	38.000	1,5	15.917	23.875	2,25
	9	12.000	2,5	- 10.083	- 25.208	6,25
	10	13.000	3,5	- 9.083	- 31.792	12,25
	11	32.000	4,5	9.917	44.625	20,25
	12	41.000	5,5	18.917	104.042	30,25
Average	6,5	22.083		Σ	221.500	143

Figure 2.5.: Double Exponential Smoothing: Calculating Level and Trend

						α	0,1	β	0,2	
Data Point	Demand	Level	Trend	Forecast	Error	Absolute Error	MSE	MAD	% Error	MAPE
0		12.015	1.549							
1	8.000	13.008	1.438	13.564	5.564	5.564	30.959.237,34	5.564	69,55	69,55
2	13.000	14.301	1.409	14.445	1.445	1.445	16.524.153,32	3.505	11,12	40,33
3	23.000	16.439	1.555	15.710	- 7.290	7.290	28.732.809,71	4.767	31,70	37,46
4	34.000	19.594	1.875	17.993	- 16.007	16.007	85.604.032,58	7.577	47,08	39,86
5	10.000	20.322	1.645	21.469	11.469	11.469	94.788.912,25	8.355	114,69	54,83
6	18.000	21.570	1.566	21.967	3.967	3.967	81.613.688,31	7.624	22,04	49,36
7	23.000	23.123	1.563	23.136	136	136	69.957.245,79	6.554	0,59	42,39
8	38.000	26.017	1.830	24.686	- 13.314	13.314	83.370.484,37	7.399	35,04	41,48
9	12.000	26.262	1.513	27.847	15.847	15.847	102.009.888,27	8.338	132,06	51,54
10	13.000	26.297	1.217	27.775	14.775	14.775	113.638.498,20	8.981	113,65	57,75
11	32.000	27.963	1.307	27.514	- 4.486	4.486	105.136.814,69	8.573	14,02	53,78
12	41.000	30.443	1.541	29.270	- 11.730	11.730	107.841.791,89	8.836	28,61	51,68

Figure 2.6.: Double Exponential Smoothing Example

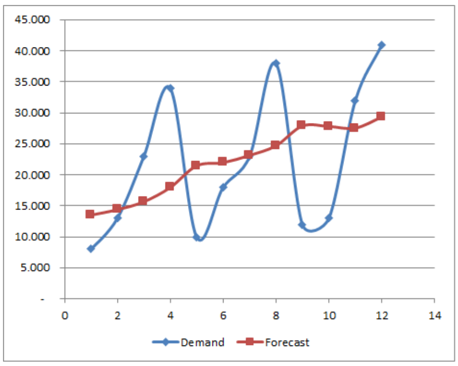

Figure 2.7.: Double Exponential Smoothing Graphical Representation

points we will need to consider the seasonality factor which we will discuss now with Triple Exponential Smoothing.

2.3.9. Triple Exponential Smoothing

To forecast more than two points we will apply exponential smoothing to a new element called *season* along with the level and the trend. Triple exponential smoothing applies exponential smoothing thrice.

If a series repeats itself in regular intervals then we call that interval the *season* and the series is referred to as *seasonal*. Triple exponential smoothing works only on seasonal series of data points. The number of data points after which a season begins is called the *season length* L. Along with level and trend we will now start to consider the seasonal component s for every data point in a season, that repeats itself at the same offset into the season.

We apply the smoothing across the different seasons by introducing a *seasonality smoothing coefficient* γ: the first point into the season will be smoothed exponentially with the first point of the last season, with first point of the two earlier seasons. We have the following 4 equations:

$$\ell_x = \alpha(y_x - s_{x-L}) + (1-\alpha)(\ell_{x-1} + b_{x-1}) \tag{2.14}$$

$$b_x = \beta(\ell_x - \ell_{x-1}) + (1-\beta)b_{x-1} \tag{2.15}$$

$$s_x = \gamma(y_x - \ell_x) + (1-\gamma)s_{x-L} \tag{2.16}$$

$$\hat{y}_{x+m} = \ell + mb_x + s_{x-L+1+(m-1)modL} \tag{2.17}$$

We thus can forecast any number of data points m into the future. The index of the seasonal component is the offset into the list of seasonal components from the last set of observed data. To have the best results, the three coefficients α, β and γ are experimentally calculated by running the process for different values to get the smallest Sum of Squared Errors (SSE), the process indicated earlier as *fitting*.

We will continue with our example this time adding the calculation for the seasonality factor. We can notice that there is a certain cyclic behaviour after time points. The first step is the calculation of level and trend, for which we use the demand without the seasonal factor. We then calculate the seasonality factor as shown in figure 2.8. The first demand without seasonal factor is calculated as $(+8000 + 10000 + ((13000 + 23000 + 34000) * 2))/8 = 19750$. This is repeated.

	Data Point	Demand	Demand without Seasonality	Linear regression on demand without seasonality				Demand without seasonality calculated	Seasonality Factor
	x	y	Ax	$p=x-\mu_x$	$q=y-\mu_y$	p*q	p^2	$A_{xb}=b_0+b_1*x$	$S = A/A_{xb}$
	1	8.000						18.963	0,42
	2	13.000						19.487	0,67
	3	23.000	19.750	-3,5	- 2.094	7.328	12,25	20.010	1,15
	4	34.000	20.625	-2,5	- 1.219	3.047	6,25	20.534	1,66
	5	10.000	21.250	-1,5	- 594	891	2,25	21.058	0,47
	6	18.000	21.750	-0,5	- 94	47	0,25	21.582	0,83
	7	23.000	22.500	0,5	656	328	0,25	22.106	1,04
	8	38.000	22.125	1,5	281	422	2,25	22.629	1,68
	9	12.000	22.625	2,5	781	1.953	6,25	23.153	0,52
	10	13.000	24.125	3,5	2.281	7.984	12,25	23.677	0,55
	11	32.000							
	12	41.000							
Average	6,5	22.083	21.844		Σ	22.000	42		
						Trend (b_1)	**524**	$\Sigma(x-\mu_x)*(y-\mu_y)$	
								$(x-\mu_x)^2$	
						Level (b_0)	**18.439**	$\mu_y-b_1\mu_x$	

Figure 2.8.: Triple Exponential Smoothing Calculations

						α	0,05	β	0,1	γ	0,1
Data Point	**Demand**	**Level**	**Trend**	**Seasonality**	**Forecast**	**Error**	**Absolute Error**	**MSE**	**MAD**	**% Error**	**MAPE**
					$=(F_{n-1}+T_{n-1})*S_n$	FIT-A	|E|	$=1/n\Sigma(E^2)$	=1/nΣ(|E|)	=|E|/A*100	=1/nΣ(|E|/A)
x	y	F	T	S	FIT	E	|E|	MSE	MAD		MAPE
0		18.439	524								
1	8.000	18.967	524	0,42	7.964	- 36	36	1.269,14	36	0,45	0,45
2	13.000	19.487	524	0,67	13.059	59	59	2.384,08	47	0,46	0,45
3	23.000	20.010	524	1,15	23.012	12	12	1.639,39	36	0,05	0,32
4	34.000	20.531	523	1,66	34.086	86	86	3.087,23	48	0,25	0,30
5	10.000	21.192	537	0,42	8.847	- 1.153	1.153	268.468,07	269	11,53	2,55
6	18.000	21.987	563	0,67	14.552	- 3.448	3.448	2.204.850,35	799	19,15	5,32
7	23.000	22.422	550	1,15	25.931	2.931	2.931	3.116.821,06	1.104	12,74	6,38
8	38.000	22.969	550	1,66	38.125	125	125	2.729.166,87	981	0,33	5,62
9	12.000	23.753	573	0,43	10.003	- 1.997	1.997	2.868.819,20	1.094	16,64	6,84
10	13.000	24.059	547	0,68	16.654	3.654	3.654	3.917.126,39	1.350	28,11	8,97
11	32.000	24.782	564	1,14	27.990	- 4.010	4.010	5.022.827,04	1.592	12,53	9,29
12	41.000	25.315	561	1,66	42.052	1.052	1.052	4.696.502,71	1.547	2,57	8,73
13				0,43							
14				0,67							
15				1,15							
16				1,66							

Figure 2.9.: Triple Exponential Smoothing

From this we elaborate the triple exponential smoothing as in figure 2.9 with $\alpha = 0.05$, $\beta = 0.1$ and $\gamma = 0.1$. The graphical form is as in figure 2.10.

The triple exponential smoothing is also called *Holt-Winter method* [85] [63] as it was first suggested by Holt's student, Peter Winters, in 1960 after reading a signal processing book from the 1940s on exponential smoothing[101].

There is also an adaptation of this method called *adaptive smoothing* where the smoothing factors are regularly reviewed at the conclusion of each forecasting period to determine the exact value that would have determined the perfect forecasting for the previous period, thereby including also a managerial decision in this approach. More sophisticated form of adaptive smoothing also includes an automatic tracking signal to monitor the error. When this signal is tripped because of excessive error the factors are automatically increased to

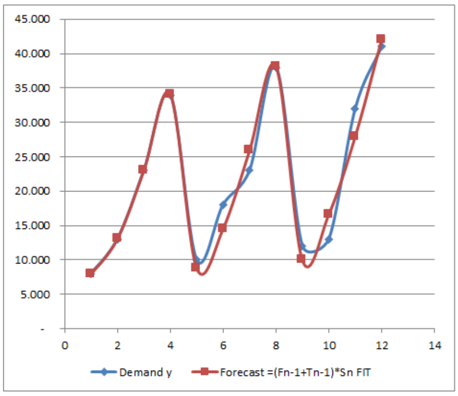

Figure 2.10.: Triple Exponential Smoothing Graphical Representation

make the forecasting more responsive to smoothing recent periods[19].

The main limitation of this approach is the lack of quick response to change since it takes numerous periods for the model to identify changes in the patterns and the forecast to respond to the pattern changes. Getting the correct values for the smoothing coefficients also is non-trivial.

2.4. Conclusion

The scope of this chapter was to present the main techniques used in demand planning, which is one of the fundamental starting point for all further discussions. The demand planning is the direct input to the Master Production Schedule (MPS) discussed in chapter 3.

Even though the quantitative methods are really elaborate and technically correct and useful, they are not often useful in real world situations as they are. When the long term and medium term planning are done, the qualitative methods are used almost always. This is also enhanced by the fact that the sales staff does not often have the technical preparation to approach the demand planning in a quantitative manner using the techniques elaborated in this chapter. Moreover the textbook presentation of these methods are too academic to be implemented in real world situations. The example that we illustrated in this chapter was implemented in excel in a comprehensive manner. Even then we need to empirically derive the level, trend and seasonality factor, which needs a certain mathematical and technical preparation.

The software solutions available on market also require a serious background training. In big companies there is the position of demand planner who mixes these methods along with software solutions, ready made or custom made, to produce the demand forecast. For any given forecast a 75% or more accuracy with respect to the real data is a considered an excellent result.

The main contribution of this chapter to the current thesis is the first glimpse of a system approach to supply chain. The demand could be considered as a vector with rows corresponding to the single time points. This vector evolves over time. We can think of the whole demand planning as a system, with the past demand as the input and future demand forecasts as the output and thus model the system. The algorithms used are the system equations, α, β and γ, the system variables. This system is dynamic, time-variant and could be linear or even more complex. In the next chapters we will see how this demand gets progressively transmitted to the demand or requirement for finished goods (MPS) and for the components (MRP), which is discussed in chapter 4.

Chapter 3.

Inventory Planning: Master Production Schedule

The demand we discussed in the previous chapter is often done for product families in an aggregate manner. This has to be brought down to the level of individual items where we discuss the individual demands for specific items. This is done in Master Production Schedule (MPS). This chapter analyzes this core process to any inventory and production planning. By production planning we mean the process of connecting this demand to a practical and feasible production plan. We can consider the whole process of production planning as a system as shown in figure 3.1. Master Production Schedule (MPS) is the first of the two core processes along with Materials Requirements Planning (MRP) that make up this system.

The end results required from this system are work orders, purchase orders and material plans. The inputs are demand forecast of the sales, already available customer orders, current stock situation as given by inventory records and Bill of Materials which details how each product is composed. There will also be many other system variables required for the correct functioning of the system, which we will see when we dig into details. In this chapter we will discuss the classic approach to Master Production Schedule and then take a systems approach to MPS calculation to then arrive at a new multidimensional matrix

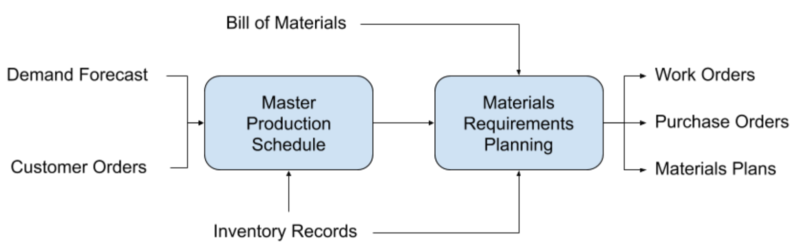

Figure 3.1.: System view of Materials Requirements Planning

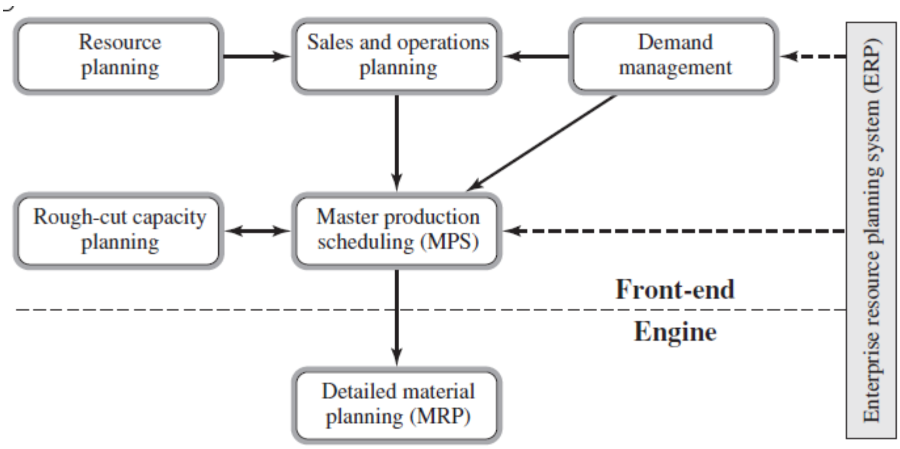

Figure 3.2.: MPS and other supply chain processes

approach to calculate MPS in an rapid and efficient algorithm.

3.1. Master Production Schedule

Master Production Schedule (MPS) is a statement of the volume and timing of the end products to be made. A stable and well defined MPS translates into stable component schedules and improved performance in the plant. MPS can be considered as a statement of planned future output.

MPS should not be confused with forecasts. Forecast is an important input to MPS. Along with forecast, MPS takes into account the capacity constraints, production costs, minimum batch sizes as well as the sales and operations plan. So MPS can propose large batch of a product while the forecast proposes a smaller one. There could also be the case where a forecast is deliberately left unmet or produced in advance with respect to the actual time of demand forecasted. The figure 3.2 [3] shows how MPS is connected to the other supply chain processes.

Master Production Schedule precisely defines the required quantity per period for each finished product to sell. This requirements derived from demand planning is matched with the current inventory (and inventory policies) and already existing production and purchasing plans to generate new planned orders and projected inventories of these products. Normally the time period is identical to that used for MRP calculations.

We can think of MPS calculation as a system in itself as shown in figure 3.3. The inputs to this system are the Demand Forecast, Allocated Customer Orders, Reserved Customer Orders and Unplanned Customer Orders along

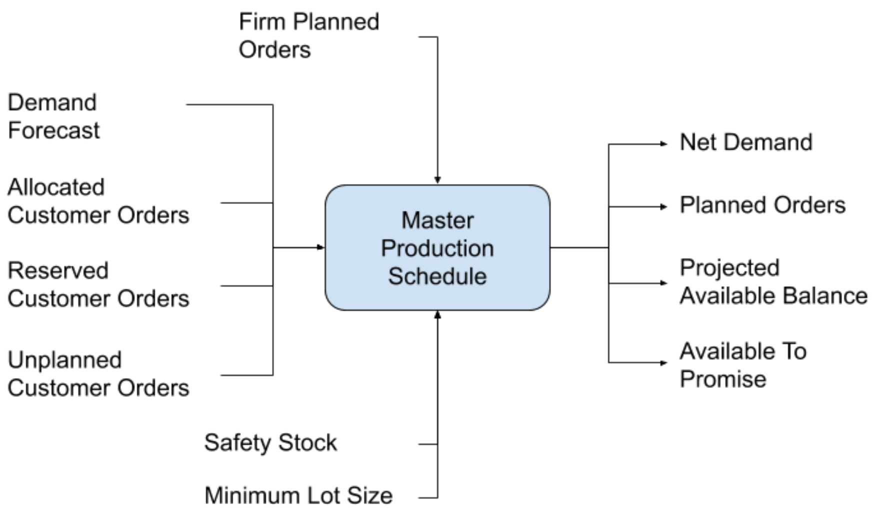

Figure 3.3.: MPS Matrix System

with Safety Stock and Minimum Lot Size. The system will calculate the Net Demand, Firm Planned Orders, Planned Order, Projected Available Balance and Available to Promise.

There are also two concepts on time which are used in MPS calculations, which we will consider as system variables, the concept of *time fence* and *planning horizon.*

3.1.1. Time Fence

Time Fence refers to the frozen period where no changes are allowed[3]. There are many different time fences:

- *Forecast time fence* - it's a period where only orders are used when calculating Net demand;
- *Reservation time fence* - only firm customer orders can be accepted for allocation during this period;
- *MPS time fence* - within this time, MPS stay firm;
- *Available to Promise time fence* - no additional orders may be accepted in this period.

3.1.2. Planning Horizon

The Planning Horizon is the time from the current date to some date in the future. It should be long enough to avoid problems with scheduling. Typically Planning Horizon should be longer than the cumulative lead time for the item to avoid unnecessary calculations.

3.2. MPS Table

In almost all modern ERP systems, all the information regarding MPS including the inputs and outputs are presented in a tabular form referred to as MPS table. Every item under MPS has its own separate table. Even though the data are stored separately in different database tables, they are jointly presented in this tabular form.

A typical MPS table contains the following data:

- *Demand forecast*: Customer demand forecast for the item, which is often integrated with internal demand forecasting as discussed in chapter 2.
- *Allocated Customer Orders (ACO)*: Orders already confirmed by the customer.
- *Reserved Customer Orders (RCO)*: Orders placed to book materials and production lines by management decision, but not sure if this will really be confirmed. This can be considered as a forced input to the system.
- *Unplanned Customer Orders (UCO)*: Unplanned (unexpected) orders which were not included in the forecast calculations, which the customers sent exceeding their forecasts.
- *Planned Orders (PO)*: Manufacturing order automatically calculated by the system.
- *Firm Planned Orders (FPO)*: Manufacturing orders which has already been released. They cannot be changed by the system - each change to firm order requires manual intervention as it could potentially create problems in shop floor. Firm orders may already be in production.
- *Projected Available Balance (PAB)*: Projected number of available items.
- *Available to promise (ATP)*: The maximum stock of the item available in any time bucket against which sales orders can be placed by the customers without creating immediate problems. If the sales promises above that quantity, the company will not be able to keep its promise and negotiation will be needed to check the possibility of adjusting MPS to satisfy this

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				40	50	10	35	20	30	30	40
Allocated Customer Orders		45	55	10	5	25	35	15			
Reserved Customer Orders				20	10	20	10				
Unplanned Customer Orders			20		20			20			
Net Demand	0	45	75	40	50	45	45	35	30	30	40
Firm Planned Orders		10									
Planned Order		20	80	40	60	40	40	40	20	40	40
Projected Available Balance	55	40	45	45	55	50	45	50	40	50	50
Available to Promise		30	25	10	45	0	0	25	20	40	40
Safety Stock	40	40	40	40	40	40	40	40	40	40	40
Minimum Lot Size	20	20	20	20	20	20	20	20	20	20	20

Figure 3.4.: Example of an MPS Table

extra demand. The logic for calculating ATP could be discrete , where the first period and every order after it are considered independent from a planning view. We can also use a cumulative approach and carry the units that we can promise from one batch forward to the next.

Many different approaches can be used in a cumulative manner to meet demand. In *Level strategy* production is held constant. In *Chase strategy* the MPS is chasing the demand, adjusting resources to match the demand. Often a mixed strategy is used which integrates both together[59].

In figure 3.4 we have an example of an MPS table. Safety stock the extra stock or inventory held in order to reduce the risk of the item being out of stock. It acts as a buffer in case the sales of an item are greater than planned and/or the supplier is unable to deliver additional units at the expected time. Minimum Lot Size is the minimum multiple in which an item can be produced (or sold). In MPS calculations the system considers this quantity while proposing planned orders. The planned order will be round a number up to nearest multiple of the minimum lot size.

3.3. Master Production Schedule Calculation

In this section we will discuss the classical approach used commonly in MPS calculations, implemented by most of the current algorithms. Every company decides on when to run the MPS calculations depending on the specific manufacturing model followed. Normally at least once a week this calculation should be done to have reliable planning data. If there are any substantial changes in order intake and of quantities starts exceeding the forecast, it is highly advised to execute MPS, referred commonly as *MPS run*.

The starting points for MPS calculation are Demand Forecast, customer orders and inventory records. For every item on which MPS calculation is done, we make an MPS record. This could be in a database or even in an Excel file.

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				40	50	10	35	20	30	30	40
Allocated Customer Orders		45	55	10	5	25	35	15			
Reserved Customer Orders				20	10	20	10				
Unplanned Customer Orders			20		20			20			
Net Demand											
Firm Planned Orders											
Planned Order											
Projected Available Balance											
Available to Promise											
Safety Stock											
Minimum Lot Size											

Figure 3.5.: MPS Calculation Example: Demand and customer orders

Two key considerations in setting up the MPS are the size of *time buckets* and the *planning horizons*. A *time bucket* is the unit of time for which the schedule is constructed and is typically daily or weekly. The *planning horizon* is how far to plan forward, and is determined by how far ahead demand is known and by the lead times. Planning horizon could stem from the time period for which useful data is available or from a management decision or from a mix of both. The number of columns in the MPS table will be equal to the number of time buckets in the planning horizon. The minimum rows required for this table are:

- Demand Forecast
- Allocated Customer Orders
- Reserved Customer Orders
- Unplanned Customer Orders
- Net Demand
- Firm Planned Orders
- Planned Order
- Projected Available Balance
- Available To Promise

For a better understanding of this process we will add another couple of rows for Safety Stock and for Minimum Lot Size.

Let us explain the how this calculation works with the help of an example. Let us consider a finished item for which we assume the demand forecasted, allocated, reserved and planned customer orders as given in figure 3.5 for the next 10 weeks. This means our time bucket is weekly and our planning horizon is 10 weeks. We assume a 2 week MPS time fence. Week 0 is the current week.

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				40	50	10	35	20	30	30	40
Allocated Customer Orders		45	55	10	5	25	35	15			
Reserved Customer Orders				20	10	20	10				
Unplanned Customer Orders			20		20			20			
Net Demand	0	45	75	40	50	45	45	35	30	30	40
Firm Planned Orders											
Planned Order											
Projected Available Balance											
Available to Promise											
Safety Stock											
Minimum Lot Size											

Figure 3.6.: Net Demand Calculation

In demand time fence we do not have any forecasted demand. The forecast time fence is 3 time buckets and so the forecast starts on the fourth time bucket as shown in figure 3.5 and extends all along to the planning horizon. From week 1 we have allocated customer orders and there are some reserved customer orders from week 3 to 6, let us assume for some management decision. There are also some unplanned customer orders in week 2, 4 and 7.

To calculate the Net Demand, we use the equation 3.1

$$ND[t] = max(F[t], (ACO[t] + RCO[t] + UCO[t])) \tag{3.1}$$

where $ND[t]$ is the Net Demand at any time bucket i, $F[t]$ the Demand Forecast of that time bucket, $ACO[t]$,$RCO[t]$ and $UCO[t]$ are Allocated, Reserved and Unplanned Customer Orders. Our hope is that during a period we will eventually sell up to the forecast, and, if this proves to be true we want to reflect the additional sales in projecting our inventory balance. This is often referred to as *consuming the forecast*.

In figure 3.6 we see this calculation. In time bucket 0,1 and 2 we are in the forecast time fence and so the Net Demand is the sum of Allocated, Reserved and Unplanned Customer Orders. In time bucket 3 we see that the maximum among Demand Forecast (40) and Customer Orders (30) is 40, which is the Net Demand and so on.

To calculate the Planned Orders, along with the Net Demand, we should consider other data. First of all is the current inventory for the first time bucket and Planned Available Balance for the successive time buckets. Firm Orders should be taken into account along with the Safety Stock requirements and Minimum Lot Sizes. It is calculated as in equation 3.2.

$$PO[t] = SS[t] + ND[t] - PAB[t-1] - FP[t] \tag{3.2}$$

where $PO[t]$ are the Planned Orders calculated for the time bucket t, $SS[t]$, $ND[t]$, $FP[t]$ Safety Stock, Net Demand and Firm Planned Orders respectively for that time bucket. $PAB[t-1]$ for the first time bucket is the available

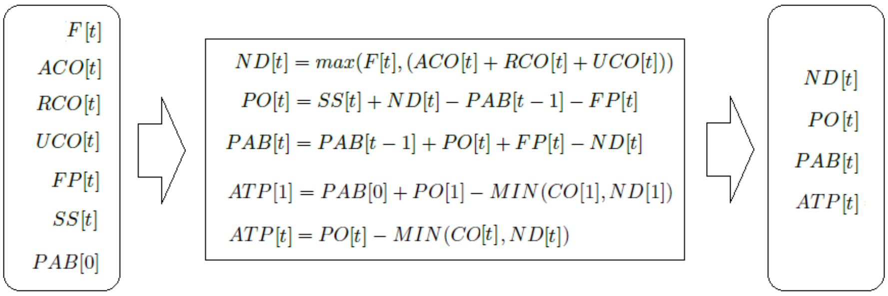

Figure 3.7.: MPS System Equations

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				40	50	10	35	20	30	30	40
Allocated Customer Orders		45	55	10	5	25	35	15			
Reserved Customer Orders				20	10	20	10				
Unplanned Customer Orders			20		20			20			
Net Demand	0	45	75	40	50	45	45	35	30	30	40
Firm Planned Orders		10									
Planned Order		20	80	40	60	40	40	40	20	40	40
Projected Available Balance	55	40	45	45	55	50	45	50	40	50	50
Available to Promise		30	25	10	45	0	0	25	20	40	40
Safety Stock	40	40	40	40	40	40	40	40	40	40	40
Minimum Lot Size	20	20	20	20	20	20	20	20	20	20	20
	MPS Time Fence										

Figure 3.8.: MPS Calculation

inventory where as for the successive time buckets are the Planned Available Balance of the previous period. Planned Available Balance can be calculated as in equation 3.3.

$$PAB[t] = PAB[t-1] + PO[t] + FP[t] - ND[t] \tag{3.3}$$

Available to Promise can be calculated starting from the first time bucket as in equations 3.4 and 3.5:

$$ATP[1] = PAB[0] + PO[1] - MIN(CO[1], ND[1]) \tag{3.4}$$

$$ATP[t] = PO[t] - MIN(CO[t], ND[t]) \tag{3.5}$$

so we can refine our system as in figure 3.7 with the inputs, outputs and system equations.

Continuing with our example, we will consider a Safety Stock of 40 items in all time buckets and a minimum lot size of 20 items as shown in the last two rows of figure 3.8.

The Planned Order of time bucket 1 is calculated using equation 3.2 as the sum of Safety Stock (40) and Net Demand (45) at time bucket 1 minus the Firm

Planned Order (10) at the same time bucket minus the Projected Available Balance at the preceding time bucket(55) giving $40 + 45 - 10 - 55 = 20$. In case this value is not a multiple of the Minimum Lot Size then we round it up to its next multiple which is greater than the calculated Planned Order. If this value is negative, we will replace it with 0 since negative values does not make any sense.

The Projected Available Balance of time bucket 1 is calculated using equation 3.3 as the sum of Projected Available Balance (inventory available) at time bucket 0 (55), Planned Orders (20) and Firm Planned Orders (10) at time bucket 1 minus the Net Demand (45) giving $55 + 10 + 20 - 45 = 40$.

For the time bucket 1, the Available To Promise is the sum of Projected Available Balance of the first time bucket(55) and the Planned Orders of the time bucket 1 (20) minus the minimum among the sum of Allocated Customer Orders (45+0) and Net Demand (45) giving $55+20-min(40+2, 45) = 75-45 = 30$. For the time bucket 2 on wards, the Available To Promise is the sum of Planned Orders of the time bucket, for example 4, (60) minus the minimum among the sum of Allocated Customer Orders $(10 + 5)$ and Net Demand (50) giving $40 - min(40, 30) = 60 - 15 = 45$. We can also have an alternative calculation including the Unplanned Customer Orders for a better calculation of ATP, summing it along with the Allocated and Reserved Customer Order Calculation.

The use of both the Projected Available Balance and the Available To Promise row is the key to effective Master Production Scheduling. Using ATP to book orders means that it is unlikely that a customer promise will be made that cannot be kept. This could mean some orders must be booked at the end of a planning horizon along with creating an additional MPS order. Since actual orders are booked (and reflected in the order row), or anticipated (in the forecast row), or shipped (as reflected in the Projected Available Balance), the Available To Promise row provides a signal for the creation of new MPS orders.

3.4. Matrix Approach to Master Production Schedule

One of the main drawbacks of approach described in the previous section is the time required to calculate the schedule. As we saw earlier, it is done one by one item per item. In this chapter we will see an alternative approach to MPS calculation using matrices and matrix manipulations. The matrix perspective is fundamental to this and in further development of this new concept since the material requirements planning (MRP) discusses in the next chapter and the capacity and inventory planning mentioned in the coming chapters are developed incrementally on this concept.

The tabular form of MPS table suggests that instead of iterating over the list of all items and then along the time axis, it could be better handled by matrix manipulation algorithms. These algorithms are way more efficient than loops and can handle large amounts of data simultaneously. The second step is to find a suitable data representation. The final step is to use this representation along with the algorithm with some adaptations to make the process more efficient.

The classical MPS calculation is done item per item along the planning horizon. This means iterating along two axes: one for item and other for time into the future. The proposed method consists of elimination of the item axis and move along just the time axis with a matrix representing all items at any given point of time. Instead of the MPS Table for single items we will have a MPS Matrix.

3.5. MPS Matrix for Single Item

The first idea is to make a set of matrices for all the entries of the MPS table. We will start with 11 matrices for the 11 entries for any single item.

The next step is to fill the matrices Net Demand, Planned Orders, Planned Available Balance and Available To Promise with zeros as in equations 3.6 to 3.9. we take $t_h + 1$, where t is the planning horizon. We will consider 0 as the current time slot and all the rest of time buckets from 1 to t as the time horizon in which MPS calculation will be done. To facilitate later discussions, we will be using mostly Matlab representation.

$$ND = zeros(1, t_h + 1); \tag{3.6}$$

$$PO = zeros(1, t_h + 1); \tag{3.7}$$

$$PAB = zeros(1, t_h + 1); \tag{3.8}$$

$$ATP = zeros(1, t_h + 1); \tag{3.9}$$

The first matrix we will calculate is the Net Demand ND using equation 3.1, this time in matrix representation as in equation 3.10. We will also remove the negative values as they make no sense.

$$ND = max(DEMAND, (ACO + RCO + UCO)); \tag{3.10}$$

Once we have the Net Demand, we can proceed to calculate the Planned Order matrix PO as in equation 3.2, which in matrix form is given by equation 3.11. We will loop for all the time buckets. This loop is not avoidable since we consider the previous values calculated at the previous step for the calculations in the current step. We will also remove the negative values since they also make no sense.

$$PO(t) = SS(t) + ND(t) - FPO(t) - PAB(t-1); \tag{3.11}$$

The Planned orders above does not consider the Minimum Lot Size as given in MLS matrix. So in every step of the loop we will round up the planned order to the next multiple of the Minimum Lot Size. The successive Planned Orders will consider this additional quantity and recalculate the new Planned Order for the successive time buckets.

We first calculate the nearest rounded up multiple of PO matrix and then multiply it with a diagonal matrix made up of the value of MLS matrix in the diagonals as in equation 3.12. We will be using the functions $ciel$, $bsxfun$ and $rdivide$ provided by Matlab.

- $Y = ceil(X)$ rounds each element of X to the nearest integer greater than or equal to that element.
- $C = bsxfun(fun, A, B)$ applies the element-wise binary operation specified by the function handle fun to arrays A and B.
- $x = rdivide(A, B)$ divides each element of A by the corresponding element of B.

$$PO = diag(MLS) * ceil(bsxfun(@rdivide, PO, MLS));; \tag{3.12}$$

Finally we calculate the Planned Available Balance matrix as in equation 3.13 and Available To Promise matrix as in equation 3.14 for the time buckets inside the MPS time fence and 3.15 for all others.

$$PAB(t) = PAB(t-1) + FPO(t) + PO(t) - ND(t); \tag{3.13}$$

$$ATP(t) = PO(t) - min(ND(t), ACO(t) + RCO(t)); \tag{3.14}$$

$$ATP(t) = PAB(t-1) + PO(t) - min(ND(t), ACO(t) + RCO(t)); \tag{3.15}$$

Let us now use the same MPS table discussed earlier and apply this new method. As input we will have the following matrices:

```
DEM = [0,0,0,40,50,10,35,20,30,30,40];
ACO = [0,45,55,10,5,25,35,15,0,0,0];
RCO = [0,0,0,20,10,20,10,0,0,0,0];
UCO = [0,0,20,0,20,0,0,20,0,0,0];
FPO = [0,10,0,0,0,0,0,0,0,0,0];
MLS = [20,20,20,20,20,20,20,20,20,20,20];
SS  = [40,40,40,40,40,40,40,40,40,40,40];
INV  = [55];
```

The full algorithm is as follows:

```
% Input matrices
DEM = [0,0,0,40,50,10,35,20,30,30,40];      % Forecast
ACO = [0,45,55,10,5,25,35,15,0,0,0];        % ACO
RCO = [0,0,0,20,10,20,10,0,0,0,0];          % RCO
UCO = [0,0,20,0,20,0,0,20,0,0,0];           % UCO
FPO = [0,10,0,0,0,0,0,0,0,0,0];             % FPO
MLS = [20,20,20,20,20,20,20,20,20,20,20];   % MLS
SS  = [40,40,40,40,40,40,40,40,40,40,40];   % SS
INV  = [55];                                % Inventory

% System Parameters
TIMEFENCE = 2;
PLANNING_HORIZON =10;

% Initialization of output matrices
ND = zeros(1,PLANNING_HORIZON+1);
PO = zeros(1,PLANNING_HORIZON+1);
PAB = zeros(1,PLANNING_HORIZON+1);
ATP = zeros(1,PLANNING_HORIZON+1);

% Calculating Net Demand
ND = max(DEM,(ACO+RCO+UCO));
ND(ND<0)=0;

% PAB(1) is initial inventory
PAB(1) = INV;

for t = 2:PLANNING_HORIZON+1

   % Calculating Planned Orders
   PO(t) = SS(t) + ND(t)-  FPO(t) - PAB(t-1);

   % Removing negative values
   TMP = PO(t);
   TMP(TMP<0)=0;
```

```
    PO(t)=TMP;

    % Rounding POs to multiples of MLS
    TMP = ceil(bsxfun(@rdivide,PO,MULTIPLE(t)));
    PO = diag(MULTIPLE(t)) * TMP;

    % Calculating Planned Available Balance
    PAB(t) = PAB(t-1) + FPO(t) + PO(t) - ND(t);

    % Calculating Available To Promise
    if t <= TIMEFENCE
     ATP(t) = PAB(t-1) + PO(t)-min( ND(t),  ACO(t) + RCO(t));
    else
     ATP(t)  =  PO(t) -min( ND(t),   ACO(t) + RCO(t));
    end

    % Removing negative values
    TMP = ATP(t);
    TMP(TMP<0)=0;
    ATP(t)=TMP;
end
```

and we get the result as follows, which is exactly the same as that of the MPS table from the earlier example.

```
ND  = [0,45,75,40,50,45,45,35,30,30,40]
PO  = [0,20,80,40,60,40,40,40,20,40,40]
PAB = [55,40,45,45,55,50,45,50,40,50,50]
ATP = [0,30,25,10,45, 0, 0,25,20,40,40]
```

The main drawback with this approach is that we have 11 matrices per item and with the high number of parts present in any company this quickly becomes an inefficient way of handling the problem. The next approach that we propose is to add an extra dimension to the matrix by creating a 3 dimensional matrix where the first dimension (rows) will be for single part numbers, the second dimension for the time buckets and the 11 matrices are stacked into the third dimension.

3.6. Multidimensional MPS Matrix

In the proposed method all the items for which MPS calculations will be done are represented in a single matrix with items along one axis, planning horizon along the other and the various information about the item on the third axis. On this third axis, continuing the earlier discussions, we will be representing

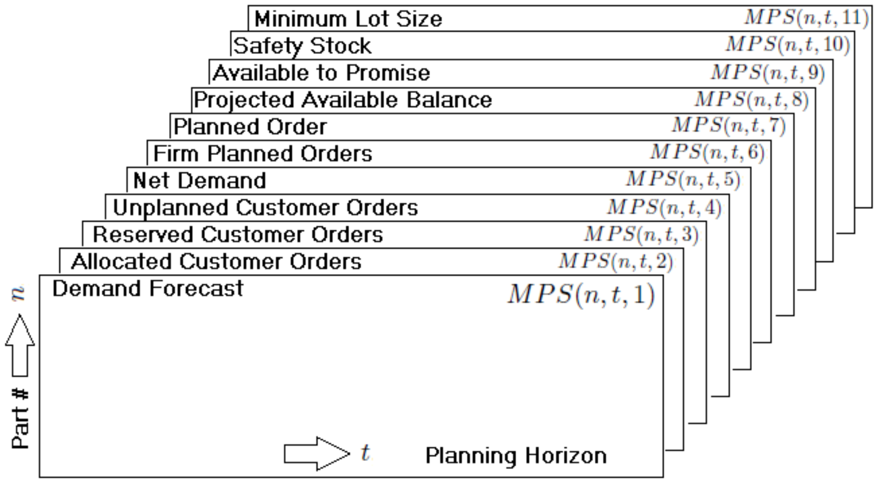

Figure 3.9.: MPS Matrix

the 11 matrices representing the 11 rows of the MPS table as shown in figure 3.9. This will be our data structure.

We will call this matrix **MPS Matrix** $MPS(n,t,11)$ of dimension $n \times t \times 11$ representing n part numbers for a time horizon t and with 11 data sets:

1. $MPS(n,t,1)$: Demand Forecast
2. $MPS(n,t,2)$: Allocated Customer Orders
3. $MPS(n,t,3)$: Reserved Customer Orders
4. $MPS(n,t,4)$: Unplanned Customer Orders
5. $MPS(n,t,5)$: Net Demand
6. $MPS(n,t,6)$: Firm Planned Orders
7. $MPS(n,t,7)$: Planned Orders
8. $MPS(n,t,8)$: Planned Available Balance
9. $MPS(n,t,9)$: Available to Promise
10. $MPS(n,t,10)$: Safety Stock
11. $MPS(n,t,11)$: Minimum Lot Size

To start with, we will have the MPS Matrix with $MPS(n,t,1)$, $MPS(n,t,2)$, $MPS(n,t,3)$, $MPS(n,t,4)$ already filled up with data from the demand planning, Allocated Customer Orders, Reserved Customer Orders and Unplanned

Customer orders respectively. In the presentation of the algorithm instead of using pseudo-code, we will be using Matlab syntax to better explain the method in a clear manner.

We will also consider the three system variables :

- $TIME\_FENCE(t_f)$
- $PLANNING\_HORIZON(t_h)$
- $INVENTORY(I)$

where $TIME\_FENCE(t_f)$ is the frozen time fence, $PLANNING\_HORIZON(t_h)$ the planning horizon and $INVENTORY(I)$ the column vector which gives the current inventory for the parts.

We will start with Demand Forecast (matrix 1), Allocated Customer Orders (matrix 2), Reserved Customer Orders(matrix 3), Unplanned Customer Orders (matrix 4), Firm Planned Orders (matrix 6), Safety Stock (matrix 10) and Minimum Lot Size (matrix 11) already filled up with input data.

The next step is to fill the matrices where we will store the results with 0 as in equations 3.16 to 3.19.

$$MPS(:,:,5) = zeros(n, t_h + 1); \tag{3.16}$$

$$MPS(:,:,7) = zeros(n, t_h + 1); \tag{3.17}$$

$$MPS(:,:,8) = zeros(n, t_h + 1); \tag{3.18}$$

$$MPS(:,:,9) = zeros(n, t_h + 1); \tag{3.19}$$

where n is the number of parts under MPS calculation.

We will first calculate the Net Demand $MPS(n, t, 5)$ using equation 3.1, this time in matrix representation as in equation 3.20:

$$MPS(:,:,5) = max(MPS(:,:,1), MPS(:,:,2) + MPS(:,:,3) + MPS(:,:,4)); \tag{3.20}$$

Once we have the Net Demand, we proceed to calculate the Planned Orders, which will be stored in the matrix $MPS(n, t, 7)$ as in equation 3.2, which in matrix form is given in equation 3.21.

$$MPS(:,t,7) = MPS(:,t,10) + MPS(:,t,5) - MPS(:,t,6) - MPS(:,t-1,8); \tag{3.21}$$

We loop from current time bucket till the end of time horizon with t representing the time bucket under analysis. Once completed, we will replace negative values in this matrix with 0 since the negative Planned Orders does make any sense and we should not produce anything in that time bucket.

The Planned Orders above does not consider the Minimum Lot Size give in $MPS(n, t, 11)$. So in every step of the loop we will round up the Planned Order to the next multiple of the Minimum Lot Size. The successive Planned Orders will consider this additional quantity and recalculate the new Planned Orders for the successive time buckets.

We first calculate the nearest rounded up multiple of $MPS(n, t, 7)$ and then multiply it with a diagonal matrix made up of the $MPS(n, t, 11)$, where t is the current time bucket as in equations 3.22 and 3.23. The temporary variable $TMP2$ is used to split the long line of code to short understandable lines. We will be using the functions $ciel$, $bsxfun$ and $rdivide$ provided by Matlab, described in the previous section.

$$TMP2 = ceil(bsxfun(@rdivide, MPS(:, :, 7), MPS(:, t, 11))); \tag{3.22}$$

$$MPS(:, :, 7) = diag(MPS(:, t, 11)) * TMP2; \tag{3.23}$$

From Net Demand we can calculate the Planned Available Balance as in equation 3.3, which in matrix form is as in equation 3.24 where t is the current time bucket.

$$MPS(:, t, 8) = MPS(:, t-1, 8) + MPS(:, t, 6) + MPS(:, t, 7) - MPS(:, t, 5); \tag{3.24}$$

To calculate ATP, following the equations 3.4 and 3.5, for those inside the time fence we will use equation 3.25 and for the successive time buckets we will use equation 3.26.

$$MPS(:, t, 9) = MPS(:, t-1, 8) + MPS(:, t, 7) - min(MPS(:, t, 5), MPS(:, t, 2) \tag{3.25}$$

$$MPS(:, t, 9) = MPS(:, t, 7) - min(MPS(:, t, 5), MPS(:, t, 2) + MPS(:, t, 3)); \tag{3.26}$$

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				40	50	10	35	20	30	30	40
Allocated Customer Orders		45	55	10	5	25	35	15			
Reserved Customer Orders				20	10	20	10				
Unplanned Customer Orders			20		20			20			
Net Demand	0	45	75	40	50	45	45	35	30	30	40
Firm Planned Orders		10									
Planned Order		20	80	40	60	40	40	40	20	40	40
Projected Available Balance	55	40	45	45	55	50	45	50	40	50	50
Available to Promise		30	25	10	45	0	0	25	20	40	40
Safety Stock	40	40	40	40	40	40	40	40	40	40	40
Minimum Lot Size	20	20	20	20	20	20	20	20	20	20	20

MPS Time Fence

Figure 3.10.: MPS Table Part #1

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				50	80	30	10	20	50	60	60
Allocated Customer Orders		10	15	30	90		50	50	60	80	80
Reserved Customer Orders			60		60	60	60				
Unplanned Customer Orders				50							
Net Demand	0	10	75	80	150	60	110	50	60	80	80
Firm Planned Orders		40									
Planned Order		0	0	60	150	60	120	60	60	60	90
Projected Available Balance	100	130	55	35	35	35	45	55	55	35	45
Available to Promise		90	0	30	0	0	10	10	0	0	10
Safety Stock	30	30	30	30	30	30	30	30	30	30	30
Minimum Lot Size	30	30	30	30	30	30	30	30	30	30	30

MPS Time Fence

Figure 3.11.: MPS Table Part #2

3.7. Multidimensional MPS Matrix Calculation: Example

To explain the proposed method, we will consider a simple MPS consisting of five parts. For comparison there are the classical MPS tables for all the five parts in figures 3.10, 3.11, 3.12, 3.13 and 3.14. The Net Demand, Planned Orders, Planned Available Balance and Available To Promise are calculated using the traditional methods. So we have $n = 5$ and $t_h = 11$ (0 is the current time bucket and there are another 10 time buckets into the future).

To apply the matrix approach, we first create the MPS Matrix $MPS(5, 11, 1)$ filling up the individual matrices. We have

- The Demand Forecast matrix as in figure 3.15 which in the matrix form is as in figure 3.16. Each row corresponds to a distinct part number or item and each column to a time bucket, just like the MPS table.

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				10	5	8	5	2	4	1	10
Allocated Customer Orders		2	12	2	6	8			14	2	2
Reserved Customer Orders			10		4	4	4				
Unplanned Customer Orders				6							
Net Demand	0	2	22	10	10	12	5	2	14	2	10
Firm Planned Orders		2									
Planned Order		0	20	12	8	12	8	0	16	0	12
Projected Available Balance	6	6	4	6	4	4	7	5	7	5	7
Available to Promise		4	0	10	0	0	4	0	2	0	10
Safety Stock	4	4	4	4	4	4	4	4	4	4	4
Minimum Lot Size	4	4	4	4	4	4	4	4	4	4	4

MPS Time Fence

Figure 3.12.: MPS Table Part #3

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				15	10	10	5	5	10	25	5
Allocated Customer Orders		10	10	5	5	5	10		15	5	5
Reserved Customer Orders			10		10	10	10				
Unplanned Customer Orders				5							
Net Demand	0	10	20	15	15	15	20	5	15	25	5
Firm Planned Orders		15									
Planned Order		0	10	10	20	10	20	10	10	30	0
Projected Available Balance	20	25	15	10	15	10	10	15	10	15	10
Available to Promise		10	0	5	5	0	0	10	0	25	0
Safety Stock	10	10	10	10	10	10	10	10	10	10	10
Minimum Lot Size	10	10	10	10	10	10	10	10	10	10	10

MPS Time Fence

Figure 3.13.: MPS Table Part #4

Time Bucket	0	1	2	3	4	5	6	7	8	9	10
Demand Forecast				40	50	10	25	20	30	30	40
Allocated Customer Orders		45	55	10	5	25	35	15			
Reserved Customer Orders				20		20	10				
Unplanned Customer Orders			10	10					10	10	
Net Demand	0	45	65	40	50	45	45	20	30	30	40
Firm Planned Orders		40									
Planned Order		0	40	40	60	40	40	20	40	20	40
Projected Available Balance	55	50	25	25	35	30	25	25	35	25	25
Available to Promise		10	0	10	55	0	0	5	40	20	40
Safety Stock	20	20	20	20	20	20	20	20	20	20	20
Minimum Lot Size	20	20	20	20	20	20	20	20	20	20	20

MPS Time Fence

Figure 3.14.: MPS Table Part #5

Demand	0	1	2	3	4	5	6	7	8	9	10
Part # 1	0	0	0	40	50	10	35	20	30	30	40
Part # 2	0	0	0	50	80	30	10	20	50	60	60
Part # 3	0	0	0	10	5	8	5	2	4	1	10
Part # 4	0	0	0	15	10	10	5	5	10	25	5
Part # 5	0	0	0	40	50	10	25	20	30	30	40

Figure 3.15.: MPS Matrix: Demand Forecast

- The Allocated Customer matrix $MPS(5, 11, 2)$ is as shown in figure 3.17.
- The Reserved Customer matrix $MPS(5, 11, 3)$ is as shown in figure 3.17.
- The Unplanned Customer matrix $MPS(5, 11, 4)$ is as shown in figure 3.19.
- The Net Demand matrix $MPS(5, 11, 5)$ is as shown in figure 3.20. Since this matrix has to be calculated, we will start with this filled with zeros.
- The Frim Planned Orders matrix $MPS(5, 11, 6)$ is as shown in figure 3.21. The Planned Order matrix $MPS(5, 11, 7)$ is as shown in figure 3.22. Since this sub-matrix has to be calculated, we will start with this filled with zeros as initial values.
- The Planned Available Balance matrix $MPS(5, 11, 8)$ is as shown in figure 3.23. Since this matrix has to be calculated, we will start with this filled with zeros.
- The Available To Promise matrix $MPS(5, 11, 9)$ is as shown in figure 3.24. Since this matrix has to be calculated, we will start with this filled with zeros.
- The Safety Stock Level Orders matrix $MPS(5, 11, 10)$ is as shown in figure 3.25.
- The Frim Planned Orders matrix $MPS(5, 11, 11)$ is as shown in figure 3.26.

The algorithm is detailed in the Appendix B.1 as an executable mcode.

3.8. MPS Matrix Calculation: Results

Executing the above code, we get the MPS Matrix completed with the 4 sub-matrices: the Net Demand as in figure 3.27, the Planned Orders as in figure

```
MPS(:,:,1) =

     0     0     0    40    50    10    35    20    30    30    40
     0     0     0    50    80    30    10    20    50    60    60
     0     0     0    10     5     8     5     2     4     1    10
     0     0     0    15    10    10     5     5    10    25     5
     0     0     0    40    50    10    25    20    30    30    40
```

Figure 3.16.: MPS Matrix: Demand Forecast as Matlab Matrix

```
MPS(:,:,2) =

     0    45    55    10     5    25    35    15     0     0     0
     0    10    15    30    90     0    50    50    60    80    80
     0     2    12     2     6     8     0     0    14     2     2
     0    10    10     5     5     5    10     0    15     5     5
     0    45    55    10     5    25    35    15     0     0     0
```

Figure 3.17.: MPS Matrix: Allocated Customer Order Matrix

```
MPS(:,:,3) =

     0     0     0    20    10    20    10     0     0     0     0
     0     0    60     0    60    60    60     0     0     0     0
     0     0    10     0     4     4     4     0     0     0     0
     0     0    10     0    10    10    10     0     0     0     0
     0     0     0    20     0    20    10     0     0     0     0
```

Figure 3.18.: MPS Matrix: Reserved Customer Order Matrix

```
MPS(:,:,4) =

     0     0    20     0    20     0     0    20     0     0     0
     0     0     0    50     0     0     0     0     0     0     0
     0     0     0     6     0     0     0     0     0     0     0
     0     0     0     5     0     0     0     0     0     0     0
     0     0    10    10     0     0     0     0    10    10     0
```

Figure 3.19.: MPS Matrix: Unplanned Customer Order Matrix

```
MPS(:,:,5) =

     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     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     0
```

Figure 3.20.: MPS Matrix: Net Demand Matrix

```
MPS(:,:,6) =

     0    10     0     0     0     0     0     0     0     0     0
     0    40     0     0     0     0     0     0     0     0     0
     0     2     0     0     0     0     0     0     0     0     0
     0    15     0     0     0     0     0     0     0     0     0
     0    40     0     0     0     0     0     0     0     0     0
```

Figure 3.21.: MPS Matrix: Frim Planned Orders Matrix

```
MPS(:,:,7) =

     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     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     0
```

Figure 3.22.: MPS Matrix: Planned Order Matrix

```
MPS(:,:,8) =

     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     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     0
```

Figure 3.23.: MPS Matrix: Planned Available Balance Matrix

```
MPS(:,:,9) =

     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     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     0
```

Figure 3.24.: MPS Matrix: Available To Promise Matrix

```
MPS(:,:,10) =

    40    40    40    40    40    40    40    40    40    40    40
    30    30    30    30    30    30    30    30    30    30    30
     4     4     4     4     4     4     4     4     4     4     4
    10    10    10    10    10    10    10    10    10    10    10
    20    20    20    20    20    20    20    20    20    20    20
```

Figure 3.25.: MPS Matrix: Safety Stock Level Matrix

```
MPS(:,:,11) =

    20    20    20    20    20    20    20    20    20    20    20
    30    30    30    30    30    30    30    30    30    30    30
     4     4     4     4     4     4     4     4     4     4     4
    10    10    10    10    10    10    10    10    10    10    10
    20    20    20    20    20    20    20    20    20    20    20
```

Figure 3.26.: MPS Matrix: Multiple Lot Size Matrix

```
MPS(:,:,5) =

     0    45    75    40    50    45    45    35    30    30    40
     0    10    75    80   150    60   110    50    60    80    80
     0     2    22    10    10    12     5     2    14     2    10
     0    10    20    15    15    15    20     5    15    25     5
     0    45    65    40    50    45    45    20    30    30    40
```

Figure 3.27.: Net Demand Matrix - Result

```
MPS(:,:,7) =

     0    20    80    40    60    40    40    40    20    40    40
     0     0     0    60   150    60   120    60    60    60    90
     0     0    20    12     8    12     8     0    16     0    12
     0     0    10    10    20    10    20    10    10    30     0
     0     0    40    40    60    40    40    20    40    20    40
```

Figure 3.28.: Planned Order Matrix - Result

3.28, Planned Available Balance as in figure 3.29 and Available To Promise as in figure 3.30.

3.9. Conclusion

We can see that the result matches exactly with the MPS tables in figures 3.10, 3.11, 3.12, 3.13 and 3.14, validating the correctness of the process.

In our test environment we created a $1000 \times 700 \times 11$ matrix simulating a 1000 part manufacturing facility in a two year time frame, considering a daily demand projection. The whole calculation took about 9 seconds in Matlab

```
MPS(:,:,8) =

    55    40    45    45    55    50    45    50    40    50    50
   100   130    55    35    35    35    45    55    55    35    45
     6     6     4     6     4     4     7     5     7     5     7
    20    25    15    10    15    10    10    15    10    15    10
    55    50    25    25    35    30    25    25    35    25    25
```

Figure 3.29.: Planned Available Balance Matrix - Result

```
MPS(:,:,9) =

     0    30    25    10    45     0     0    25    20    40    40
     0    90     0    30     0     0    10    10     0     0    10
     0     4     0    10     0     0     4     0     2     0    10
     0    10     0     5     5     0     0    10     0    25     0
     0    10     0    10    55     0     0     5    40    20    40
```

Figure 3.30.: Available To Promise Matrix - Result

running on a Windows 10 a tablet with Intel Z8350 processor, RAM 2 GB. The same data set on a distributed AWS cloud running two traditional ERPs took little less than 8 minutes on the first and 5 minutes on the second. We do not know what exact algorithms are used in these systems (SAP and BAAN) but from the log displayed it seems that they are using the classical approach. Obviously this is the only comparison test that we could do since all the proprietary ERP systems does not reveal the algorithms they use. In the future it would be nice to do a bench marking with the algorithms that they use. The results obtained shows a heavy improvement over performance. We can thus think of a quick MPS, almost real time if needed with adequate hardware support and software optimizations.

The method uses a system approach to calculate the Master Production Schedule for the entire time horizon. Following the MPC methodology only the first time buckets are taken into serious consideration. The exact number of time buckets considered and the importance given to each depends on the company policy. The MPS time horizon is considered frozen for net demand, planned orders, planned available balance and available to promise. From the forecast time fence point of view, often taken by production schedulers, only orders are used when calculating net demand. In the reservation time fence only firm customer orders can be accepted for allocation and in available to promise time fence no additional orders may be accepted. So based on the required point of view only that many time buckets to the future are considered to a high accuracy. The later outputs are often considered indicative for mid and long term planning decisions. These fences will give the control signals to the executing part of the company where they can reliably consider these numbers to calculate the production capacity required, inventories and cash considerations.

This chapter links Demand Planning discussed in chapter 2 to Material requirement planning which will be explained in chapter 4, linking demand forecasting done at a family level to the one done at component level. This chapter on Master Production Schedule, along with presenting the process of MPS, introduced us to the matrix approach more in detail. We also saw this under a Model Predictive Control perspective and how the choice of time fence can be simplified using matrix structure and how scenario analysis and rolling back and forth in time is made possible. The matrix approach mentioned earlier in chapter 2 is developed for the first time in our discussion in detail in a practical case. This approach will be further elaborated when we discuss Material Requirement Planning (MRP) in the next chapter.

Chapter 4.

Inventory Planning: Material Requirements Planning

In this chapter we will consider the Material Requirements Planning (MRP), the second subsystem immediately following the Master Production Schedule in production and inventory planning. Material Requirements Planning is a set of calculations embedded in a system that helps to evaluate volume and time for planning and control purposes [64]. APICS (American Production and Inventory Control Society) advocated this approach in the 1970s. In the 1980s it expanded to Manufacturing Resource Planning (MRP II) to finally develop into a full fledged Enterprise Resource Planning (ERP) integrating information from all parts of the organization.

The main idea behind MRP is to produce only the quantity of the components needed for producing the required quantity of the finished goods at the required time. Continuing the discussion from the previous chapter we can consider MRP as a system in itself as shown in figure 4.1.

In this chapter we will be concentrating only on how the quantity of material required for satisfying MPS are calculated. We will be focusing on the computational aspects, without entering into the broader theoretical or management aspects, nor its detailed correlation with other systems.

4.1. Inputs for Material Requirements Planning

MRP needs three types of information to prepare the material planning for dependant demand of the components: Master Production Schedule, Bill of Materials and Inventory and Released orders. We discussed MPS in chapter 3 along with a new matrix approach.

A *Bill Of Materials (BOM)* is a list of the materials, components and subassemblies required to make a product along with the quantity needed for each one of them[24]. The Bill Of Materials could be of many types for the same product: they could define products as they are designed (engineering

BOM), ordered (sales BOM), built (manufacturing BOM) or maintained (service BOM). The different types of BOMs depends on the business needs and use for which they are intended. The Bill Of Materials specifies the relationship between the finished product (independent demand) and the components (dependent demand). MRP takes as input the information contained in the BOM to calculate the net requirement to satisfy the demand [1].

BOMs are hierarchical in nature, with the top level representing the finished product which may be a sub-part or a finished item. They can also branch down to multiple levels. Thus we can have single level as well as multilevel BOMs [66] [34]:

- *Single Level BOM*: each assembly is held once in a single level BOM and then has pointers to each single level BoM of its assemblies. This is useful only on BOMs without many sub-assemblies.
- *Intended BOM*: gives a hierarchical presentation with indentation. This representation is helpful in understanding the structure of the product, but all the parts belonging to an assembly are repeated.
- *Summarized BOM* : gives the total usage of each component or sub-assemblies required to produce the product in a single list, but results in the duplication of the product's assemblies

A BOM explosion helps to break an assembly or sub-assembly into its components or raw materials while the BOM implosion helps to link a component to an assembly, sub-assembly or products where it is used. The BOM is normally stored in a relational database and it is used in Materials Requirement Planning systems to determine exactly how many components are needed in order to produce the quantity of finished products stated in the MPS and to determine the items for which purchase or production orders must be released.

The Inventory records gives the stock of the components already present in the warehouses of the company. The stock could be in usable condition or on some kind of on-hold or reserve, for example due to some quality issues. MRP considers only usable stock. The Released Orders are already confirmed production or purchase orders for the components which will be delivered in the future according to the lead time of the item and replenishment order.

4.2. Steps in Material Requirements Planning

From the three inputs detailed in the previous section MRP performs four functions:

- Calculating gross requirements;

- Calculating net requirements netting stock, planned replenishment and lot sizes;
- Time-phase the net requirements including the lead times;
- Generate a list of requirements for individual components.

In traditional MRP calculations these operations are sequentially applied to all items at a given BOM level. The calculation is repeated at the BOM level below and continues descending level by level until arriving at the lowest level of the Bill Of Materials. MRP is this iterative approach where for each item the demand is distributed iteratively along the BOM tree calculating the sum of requirements. On all modern ERP systems the calculation of the derived demand of the next level is done by iterative database queries implemented in DBMS (Data Base Management Systems) in ERP (Enterprise Resource Planning) systems. This calculation could also be done is done by theoretically approaching it as an optimal control problem [30]

4.2.1. Calculating Gross Requirements

Gross requirement is the total quantity of components needed for producing any part present in the inventory of a company to satisfy the demands of the customer as calculated in the MPS. When a part is present in the Master Production Schedule, the gross requirements for the part is the independent demand for that part. If that part is composed of sub-assemblies then the demand of the item is transferred to the sub assemblies based on their multiplicity. The demand that the sub assemblies receive is called dependant demand since they depend on the demand of the parent item. Let us take the simple model of a factory producing one single product A, which is composed of 2 parts of B and 3 parts of C. The demand for the item A from the customers, say 100 is the independent demand. The demand for B (200 pcs) and C (300 pcs) are called dependant demand and they consider the multiplicity of 2 and 3 respectively of part B and C present in the BOM of part A. The gross requirement is the requirement for any single part after the dependant demand of all first level parts are distributed down to the last level of the BOM tree.

4.2.2. Calculating Net Requirements

Once the gross requirement is known for all items, we will take into consideration the current stock present in the inventory and the planned replenishment orders to get the net requirement. Net requirement is the difference between the gross requirement and the sum of stock and planned replenishment orders.

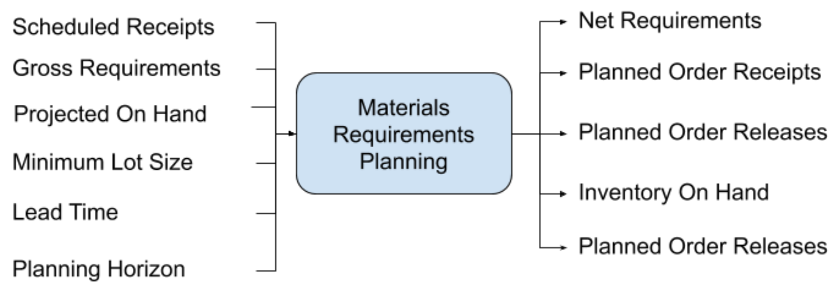

Figure 4.1.: Material Requirements Planning: System view

Net requirement refers to the demand that cannot be met with Projected On Hand or Scheduled Receipts. They must be satisfied by releasing production or purchase orders. Just as in the case of MPS, the resulting quantity is rounded up to the next multiple of the minimum lot size.

4.2.3. Considering Lead Times

Only in an ideal world the components are available as soon as we need them. In practical life there is a time difference between when an item is ordered and when it is delivered [14][29]. This can be considered as a delay in the input of the system and is referred to as lead time.

Lead times are taken into account by time phasing the demand to the past by the amount of delay so that when the time comes for its real demand the item is already available in stock. In our previous example if the item B has 1 week lead time and item C has 2 weeks lead time, to produce item A we need to order the item B a week earlier and item C two weeks earlier than the date on which the production of A is planned.

4.2.4. Calculating a list of requirement for single components

The whole purpose of MRP is to generate this list of net requirements for all single components needed to satisfy the MPS according to BOM. The production orders or purchase order if it is a purchased item, is released based on this list. We will discuss this in detail in the coming sections.

4.2.5. MRP Table

All the information regarding MRP including the inputs and outputs are normally presented in a tabular form referred to as MRP table in the classic approach to MRP calculations, used almost as it is by all current ERP systems.. Every item under MRP has its own separate table. In all ERP systems, even thought the data are stored separately in different database tables, they are jointly presented in this tabular form.

A typical MRP table contains the following data:

- *Gross Requirements*: The total quantity needed for any part, present in the inventory of a company, to satisfy the demands the parent item, which is derived from the demand of the customer during demand planning.
- *Scheduled Receipts*: Already released production and purchase orders. Some may already be in production.
- *Projected On Hand*: It gives the situation of stock based only on confirmed orders. It is obtained by subtracting Gross Requirement from the sum of initial stock at first time bucket (or previous stock from the second time bucket onwards) and scheduled receipts.
- *Net Requirements*: The requirements that cannot be met by Projected On Hand or Scheduled Receipts. They must be satisfied by releasing new production or purchase orders.
- *Planned Order Receipts*: These are the orders that are already planned in the system, computed on an earlier MRP run considering the net requirement rounded up to the multiple of Minimum Lot Size. Each execution of MPS is referred to as MPS run.
- *Planned Order Releases*: These are the new orders that are to be planned to satisfy the resulting demand considering all the scheduled and planned receipts. They are to be placed considering both the Minimum Lot Size and the Lead Time.
- *Inventory On Hand*: Is the demand subtracted from the sum of inventory at the previous time bucket, the planned receipts and the scheduled receipts.
- *Work In Process*: Refers to the number of units released to production but not yet received.
- *Minimum Lot Size*: the minimum multiple in which an item can be produced or purchased. All new orders released should be a multiple of the Minimum Lot Size.
- *Lead Time*: Delay between when the order is placed for an item (production/purchase) and when that item is received.

- *Planning Horizon*: The Planning Horizon is the time from the current date to some date in the future. It should be long enough to avoid problems with scheduling. Typically Planning Horizon should be longer than the cumulative lead time for the item.

4.3. Material Requirements Planning

4.3.1. Material Requirements Planning: Process

Let us examine how the Material Requirements Planning calculation works. We will be using the rows of the MRP table as variables in the following manner:

- Gross Requirements $GR(t)$
- Scheduled Receipts $SR(t)$
- Projected On Hand $POH(t)$
- Net Requirements $NR(t)$
- Planned Order Receipts $POR(t)$
- Planned Order Releases $POX(t)$
- Inventory On Hand $INV(t)$
- Work In Process $WIP(t)$
- Minimum Lot Size $MLS(t)$
- Lead Time $LT(t)$
- Planning Horizon t_h

These variables can be considered as time dependant. The whole calculation could be considered as a dynamic time variant system with the inputs, outputs and system equations as shown in figure 4.2.

The Gross Requirement $GR[t]$ is the total requirement of the component to produce all the parent items at time t. Since an item could be present in more than one BOM of different parent items, we should sum up the demand of the respective parent items to get the real demand of the component. The Scheduled Receipts $SR[t]$ are the already released production and purchase orders from the past that will arrive in that specific time bucket. Some may already be in production. These two are inputs to the system along with the current stock of the item $INV(t)$, Minimum Lot Size $MLS(t)$ and the production or purchase Lead Time $LT(t)$ of the item at time t form the input of the system in figure 4.2

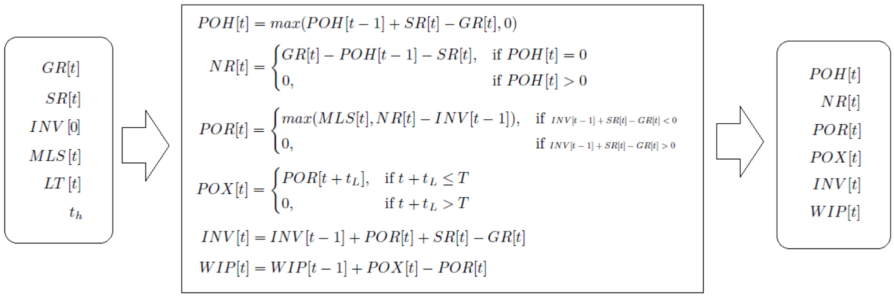

Figure 4.2.: MRP System Equations

The first output we will calculate is the Projected On Hand $POH(t)$ at time t, following equation 4.1.

$$POH[t] = max(POH[t-1] + SR[t] - GR[t], 0) \tag{4.1}$$

It gives the situation of stock based only on confirmed orders. obtained by subtracting Gross Requirement from the sum of initial stock at the first time bucket (or previous stock from the second time bucket) and scheduled receipts.

Once we have the Gross Requirement, Scheduled Receipts and Projected On Hand, we can calculate the Net Requirement $NR(t)$ at time t as in equation 4.2.

$$NR[t] = \begin{cases} GR[t] - POH[t-1] - SR[t], & \text{if } POH[t] = 0 \\ 0, & \text{if } POH[t] > 0 \end{cases} \tag{4.2}$$

We can see that at time t if the Projected On Hand $POH(t) > 0$ then we do not have any Net Demand since the Gross demand is satisfied by the inventory on hand. In case $POH(t) = 0$ then we get the Net Requirement $NR(t)$ at time t by subtracting the Projected On Hand at instance $t-1$ and Scheduled Receipts $SR(t)$ at time t from the Gross Requirement $GR(t)$ at time t.

Once we have the Net Demand we can calculate the Planned Order Receipts $POR(t)$ at time t using the equation 4.3.

$$POR[t] = \begin{cases} max(MLS[t], NR[t] - INV[t-1]), & \text{if } eq.4.4 \\ 0, & \text{if } eq.4.5 \end{cases} \tag{4.3}$$

$$INV[t-1] + SR[t] - GR[t] < 0 \tag{4.4}$$

$$INV[t-1] + SR[t] - GR[t] > 0 \tag{4.5}$$

If the difference between the Gross Requirement $GR(t)$ at time t and the sum of Inventory On Hand $INV(t-1)$, which comes from the stock at time $t-1$, and Scheduled Receipts $SR(t)$ at time t is greater than 0 then the Planned Order Receipt is 0. If it is less than 0 then the $POR(t)$ is the maximum among the Minimum Lot Size $MLS(t)$ at time t and the difference between Net Requirement $NR(t)$ at time t and stock $INV(t-1)$ at time $t-1$.

We can calculate the Planned Order Releases $POX(t)$ at time t as in equation 4.6

$$POX[t] = \begin{cases} POR[t+t_L], & \text{if } t+t_L \leq T \\ 0, & \text{if } t+t_L > T \end{cases} \tag{4.6}$$

This is a time phasing of the Planned Order Receipt $POR(t)$ according to the Lead Time of the item $LT(t)$. We consider Lead Time as time variant even though almost all of the current implementations consider a single time invariant value LT present in the item master data. We will later see that this is an important improvement when we will treat the whole MRP calculation as a dynamic data structure that evolves over time with all its variables, without anchoring to any time invariant data. If we examine in detail we can reasonably argue that the Lead Time can vary with time since it is influenced by quite a lot of factors which on their own are time variant. A lead time of 2 time buckets can become 4 due to some material shortage at the supplier for a short period and then get back to normal. A time invariant implementation is unable to manage this.

We anticipate or postpone the demand according to the value present in $LT(t)$. For example if we have $LT(t) = 1$ then Planned Order Release at time t will be the POR at time $t+1$. This means that to calculate POX we need to have the complete picture of the evolution of the system in time. For our discussions we will consider $POR(t)$ as the moment in which the item is produced, when all the components needed should be present at the production or assembly facility. We will let all the dependant demand stem from $POR(t)$ in our calculations.

At this point we can calculate the inventory progression $INV(t)$ as in equation 4.7, where the stock $INV(t)$ at time t is calculated from the stock at the previous time bucket $t-1$ summing to it the Planned Order Receipts $POR(t)$ and Scheduled Receipts $SR(t)$ at time t and then subtracting the Gross Requirement $GR(t)$.

$$INV[t] = INV[t-1] + POR[t] + SR[t] - GR[t] \tag{4.7}$$

The Work In Process $WIP(t)$ at time t is calculated as in equation 4.8, summing Planned Order Releases $POR(t)$ and subtracting Planned Order Releases

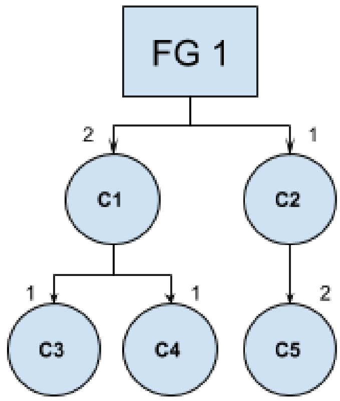

Figure 4.3.: BOM for example

	0	1	2	3	4	5	6	7	8	9	10
FG1	0	100	100	100	100	100	100	100	100	100	100

Figure 4.4.: MRP Table: Finished Good

$POR(t)$ at time t to the Work In Process at time $t-1$.

$$WIP[t] = WIP[t-1] + POX[t] - POR[t] \tag{4.8}$$

The complete system and equations are shown in figure 4.2.

4.3.2. Material Requirements Planning: Example

We will now see how the material requirements are calculated with the help of a very simple example. We will calculate the requirements for producing a single item while explaining how it works. The BOM of this item is shown in figure 4.3. The piece of $FG1$ is composed of two sub-assemblies $C1$ and $C2$. Two pieces of $C1$ and one piece of $C2$ are needed to produce a finished item of $FG1$. $C1$ is made from one piece each of $C3$ and $C4$ while $C2$ is made from two pieces of $C5$.

We will assume that the MPS for $FG1$ is done and the Planned Orders are as in figure 4.4.

For our analysis we will consider just the first 4 time buckets in order to keep the discussion short.

We see that in the current time bucket there is no demand. In the time bucket 1 there is a demand for 100 items. This means that we will need 200

C1	0	1	2	3	4	5	6	7	8	9	10
Gross Requirements	0	200	200	200	200	200	200	200	200	200	200
Scheduled Receipts	0	0	0	0	0	0	0	0	0	0	0
Projected on-Hand		0	0	0	0	0	0	0	0	0	0
Net Requirements		200	200	200	200	200	200	200	200	200	200
Planned Order Receipts		0	0	150	150	300	150	150	300	150	150
Planned Order Releases	0	0	150	150	300	150	150	300	150	150	0
Inventory on Hand	500	300	100	50	0	100	50	0	100	50	0
Work in Process		0	150	150	300	150	150	300	150	150	0
Minimum Lot Size	150	150	150	150	150	150	150	150	150	150	150
Lead Time	1	1	1	1	1	1	1	1	1	1	1

Figure 4.5.: MRP Table: C1

items of $C1$ and 100 items of $C2$ to produce 100 items of $FG1$. We will assume that there is no stock of $FG1$.

If we consider just the demand we would need 200 pieces of $C1$ and 100 pieces of $C2$. But there could be a stock of these components. We will assume to have 500 pieces of $C1$ and 50 pieces of $C2$ in stock. This means in time bucket 1 the real need comes down to no piece of $C1$ and 50 pieces of $C2$.

The complete table for C1 is shown in figure 4.5.

The Gross Requirement is the gross demand for the item to produce the parent item according to all the BOMs in which this item is present. In our example we have a single BOM. Since we need 2 pieces of $C1$ to make $FG1$, the gross demand is the PO of $FG1$ multiplied by two. There are no Scheduled Receipts.

Now we will calculate the Projected On Hand $POH(1)$ using equation 4.1. Since $max(0 + 0 - 200, 0) = 0$ we have $POH(1) = 0$.

To calculate the Net Demand we will use equation 4.2. Since we have $POH(1) = 0$ we will be using the top part of the equation and get $NR(1) = 200 - 0 - 0 = 200$.

The Planned Order Receipts $POR(1)$ can now calculated using equation 4.3. Since $INV(0) + SR(1) - GR(1) = 500 - 0 - 200 > 0$, we have the condition as in equation 4.5. So we have $POR(1) = 0$. This essentially means that we do not need any new production or replenishment since the current stock covers the requirements.

To calculate the Planned Order Releases we need to get $POR(2)$ since we have the lead time of the component $LT(1) = 1$. At the moment we can only calculate $POX(0) = POR(1) = 0$

The Inventory $INV(1)$ is calculated using the equation 4.7. So we have $INV(1) = INV[0] + POR[1] + SR[1] - GR[1] = 500 + 0 + 0 - 200 = 30$

Equation 4.8 gives the Work In Process $WIP(1) = WIP[0] + POX[1] -$

C2	0	1	2	3	4	5	6	7	8	9	10
Gross Requirements	0	100	100	100	100	100	100	100	100	100	100
Scheduled Receipts	0	0	0	0	0	0	0	0	0	0	0
Projected on-Hand		0	0	0	0	0	0	0	0	0	0
Net Requirements		100	100	100	100	100	100	100	100	100	100
Planned Order Receipts		100	100	100	100	100	100	100	100	100	100
Planned Order Releases	100	100	100	100	100	100	100	100	100	100	0
Inventory on Hand	50	50	50	50	50	50	50	50	50	50	50
Work in Process		0	0	0	0	0	0	0	0	0	-100
Minimum Lot Size	100	100	100	100	100	100	100	100	100	100	100
Lead Time	1	1	1	1	1	1	1	1	1	1	1

Figure 4.6.: MRP Table: C2

C3	0	1	2	3	4	5	6	7	8	9	10
Gross Requirements	0	0	0	150	150	300	150	150	300	150	150
Scheduled Receipts	0	0	0	0	0	0	0	0	0	0	0
Projected on-Hand		0	0	0	0	0	0	0	0	0	0
Net Requirements		0	0	150	150	300	150	150	300	150	150
Planned Order Receipts		0	0	100	150	300	150	150	300	150	150
Planned Order Releases	0	0	100	150	300	150	150	300	150	150	0
Inventory on Hand	50	50	50	0	0	0	0	0	0	0	0
Work in Process		0	100	150	300	150	150	300	150	150	0
Minimum Lot Size	1	1	1	1	1	1	1	1	1	1	1
Lead Time	1	1	1	1	1	1	1	1	1	1	1

Figure 4.7.: MRP Table: C3

$POR[1] = 0 + 0 - 0 = 0$.

In a similar manner we can calculate the MRP table for $C2$, $C3$, $C4$ and $C5$ as shown in figures 4.6, 4.7, 4.8 and 4.8 respectively.

The Gross Requirement for $C3$ and $C4$ derives from the Planned Order Releases of $C1$ and that of $C5$ from Planned Order Releases of $C2$. In time bucket 1, we see that there are no Planned Order Releases for $C1$ since all the requirement is covered by stock. $C3$ and $C4$ thus does not have any Gross Requirements.

Even though this calculation is relatively simple to understand, it requires a lot of computational effort due to looping on all items level by level and requires quite a lot of arithmetic calculations. All the current implementations of the algorithm uses relational databases, which means that there is also an additional overhead of continuous data access both in read and in write. With a high number of parts and a multilevel BOMs this calculation becomes even more time consuming and complicated forcing all current implementations to run in batch mode when no users are working on the system and then update

C4	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Gross Requirements	0	0	0	150	150	300	150	150	300	150	150
Scheduled Receipts	0	0	0	0	0	0	0	0	0	0	0
Projected on-Hand		0	0	0	0	0	0	0	0	0	0
Net Requirements		0	0	150	150	300	150	150	300	150	150
Planned Order Receipts		0	0	150	150	300	150	150	300	150	150
Planned Order Releases	0	0	150	150	300	150	150	300	150	150	0
Inventory on Hand	0	0	0	0	0	0	0	0	0	0	0
Work in Process		0	150	150	300	150	150	300	150	150	0
Minimum Lot Size	1	1	1	1	1	1	1	1	1	1	1
Lead Time	1	1	1	1	1	1	1	1	1	1	1

Figure 4.8.: MRP Table: C4

C5	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Gross Requirements	0	200	200	200	200	200	200	200	200	200	200
Scheduled Receipts	0	0	0	0	0	0	0	0	0	0	0
Projected on-Hand		0	0	0	0	0	0	0	0	0	0
Net Requirements		200	200	200	200	200	200	200	200	200	200
Planned Order Receipts		0	0	100	200	200	200	200	200	200	200
Planned Order Releases	0	0	100	200	200	200	200	200	200	200	0
Inventory on Hand	500	300	100	0	0	0	0	0	0	0	0
Work in Process		0	100	200	200	200	200	200	200	200	0
Minimum Lot Size	1	1	1	1	1	1	1	1	1	1	1
Lead Time	1	1	1	1	1	1	1	1	1	1	1

Figure 4.9.: MRP Table: C5

the MRP table. There are no known rapid methods in MRP calculations.

4.4. Multidimensional Matrix Approach to Material Requirements Planning

4.4.1. Requirements Calculation using Matrices

In this section we will present a new procedure to evaluate MRP based on a matrix approach. The use of matrices in MRP calculations is not a novelty in itself. It was first proposed by Wassily Leontief [49] and developed later by Robert Grubbstrom and his team [28]. Later on we will discuss in detail where the incremental novelty is. The MRP will be modeled as a dynamic system using time dependent matrices as data structure for storing the information normally stored in relational database. This particular data structure will allow us to use powerful matrix calculation algorithms readily available. We will use Matlab environment for all our calculations.

The starting point for this new methodology stems from the concept originally presented by Wassily Leontief in his nobel winning input-output economics theory, called Leontief's Inverse [49] [53], specifically the open model. We will use his idea of input-output matrix and then build and develop our model on top of this.

The idea is to represent the BOM by means of a matrix, which we will call BOM Matrix B, a square matrix $n \times n$ where n is the total number of items appearing in the MRP, including both the finished goods as well as the components. This matrix considers all the possible levels of the BOMs. The n items in MRP calculations appear both along the rows and along the columns. Care should be taken to keep the same order of the items along the rows and the columns.

Let us consider the following matrix B where $b(i, j)$ represents the number of items of the component at position j needed to manufacture the component at position i with $i = 1 \cdots n$ and $j = 1 \cdots n$. In Leontief's model this matrix is called input-output matrix.

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1,n} \\ b_{21} & b_{22} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{n,n} \end{bmatrix}$$

If the BOMs are correctly designed, the matrix B is either directly in upper triangular form or it is easily reducible to upper triangular form. For example if the item at row 1 needs 3 items at column 5 and 2 items at column 7 we will have $B(1,5) = 3$ and $B(1,7) = 2$. A missing upper triangularity is a clear sign of error in BOM.

Let us consider the demand for these items as the Demand Vector D

$$D = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

Let R be the Requirement Matrix

$$R = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

In the open Leontief model, $B \neq 0$ and $D \neq 0$. The next step is to calculate what we would call the Requirements matrix R as in equation 4.9. This matrix essentially tells us the *total* requirements of components (in the various columns) to make one piece of each item (in the rows) considering the whole BOM tree. This matrix is also an upper triangular $m \times m$ matrix.

$$I_n R - BR = D$$
$$(I_n - B)R = D$$

$$R = (I_n - B)^{-1} D \tag{4.9}$$

If $(I_n - B)^{-1}$ exists then it is called the Leontief Inverse [49] and this is a linear system of equations with a unique solution, and so given some final demand vector the required output can be found. Furthermore, if the principal minors of the matrix $(I_n - B)$ are all positive (known as the Hawkins–Simon condition [32]), the required output vector R is non-negative [60].

Though this concept is excellent in calculating the requirements of a flat BOM at a single instance of time, it fails to include calculation of stock already present in the warehouse, time phasing and moreover to accommodate changes in BOM.

The BOM is never static and could be thought of as a matrix evolving over time. This requirement is of the utmost importance for the MRP computation. This is also the main drawback of the earlier attempts cited earlier in repre-

```
BOM =

     0     2     1     0     0     0
     0     0     0     1     1     0
     0     0     0     0     0     2
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
```

Figure 4.10.: BOM as Input-Output Matrix

```
REQUIREMENT =

     1     2     1     2     2     2
     0     1     0     1     1     0
     0     0     1     0     0     2
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

Figure 4.11.: $(I_n - B)^{-1}$

senting the MRP calculations exclusively using matrices. We will explain this point continuing with our previous example.

Let us take the BOM in figure 4.3. We can represent this in a matrix form as shown in figure 4.10. So the elements from row 1 of this matrix b_{12} and b_{13} tells us that to make one item of $FG1$ we need 2 pieces of $C1$ and 1 piece of $C2$ and so on.

From this BOM matrix we can calculate the requirement of all components to produce one item, the matrix $(I_n - B)^{-1}$ as shown in figure 4.11.

With the demand as shown in figure 4.12, the resulting matrix $R = (I_n - B)^{-1}D$ is shown in figure 4.11. To make 100 pieces of $FG1$ we would need 200 pieces of $C1$, 100 pieces of $C2$, 200 pieces of $C3$,200 pieces of $C4$ and 200 pieces of $C5$.

Here we see the first challenge: Let us consider the current stock of the

```
DEMAND =

   100     0     0     0     0     0
```

Figure 4.12.: Demand Matrix

```
REQUEST =

   100   200   100   200   200   200
```

Figure 4.13.: Requirement Matrix

```
STOCK =

     0   500    50    50     0   500
```

Figure 4.14.: Stock Matrix

finished goods and components as shown in the matrix in figure 4.14. We have the stock of 0 pieces of $FG1$, 500 pieces of $C1$, 50 pieces of $C2$, 50 pieces of $C3$, 0 pieces of $C4$ and 500 pieces of $C5$. This stock should be subtracted from the Requirement Matrix R of figure 4.11. If we take a planning horizon with n time buckets, we have to do this stock correction on the actual requirement repeatedly. Moreover the BOM may change during those time buckets adding new components to the list, removing already existing components or changing the quantity needed for already existing components. We can think of a rapid solution of subtracting the stock from the current Requirement Matrix. But what should we do with the future?

Another challenge comes from the different lead times for different components. For example let us consider the following lead times: 1 time bucket for $FG1$, 5 time buckets for $C1$, 3 time buckets for $C2$, 2 time buckets for $C3$, 2 time buckets for $C4$ and 1 time bucket for $C5$, as shown in the Lead Time Matrix in figure 4.15. There is no way to express this using methods we discussed till now.

We also need to consider the already existing replenishment orders as shown in figure 4.16. This means that we will be receiving, because of orders placed in the past, 0 pieces of $FG1$, 100 pieces of $C1$, 10 pieces of $C2$, 10 pieces of $C3$, 0 pieces of $C4$ and 400 pieces of $C5$, which needs to be subtracted from the Requirements Matrix. The problem here is exactly the same one we already saw while discussing stock.

We can summarize the difficulties of directly using the input-output matrix

```
LT =

     1     5     3     2     2     1
```

Figure 4.15.: Lead Time Matrix

```
EXISTING =

     0   100    10    10     0    40
```

Figure 4.16.: Already Released Replenishment Matrix

as it is, as follows:

- The BOM is seldom in a static form. There can be variations in the future both in the part list of any given item as well as the quantity of parts used. This could be caused by a revision of the product structure, an improvement or simplification of the product or component, a new release or obsolescence. The formalism introduced above is not able to consider these variations.

- An inventory could be present and this should be subtracted from the demand, along with any already released replenishment orders.

- The Lead time is not taken into account. Almost no one works with full stock nor in a complete just-in-time manner. This means that there will be a time difference between when the demand is placed to a supplier and when the supplier actually delivers the part. This time difference has to be taken into account in all demand calculations [14][29].

4.4.2. Expanded Requirements Calculation using Matrices

The first solution to the above mentioned problems is to find a new data structure that can represent this time variant dynamic system in a better way. This we will obtain by adding three more matrices for lead time (`LT`), stock (`STOCK`) and already confirmed replenishment orders (`EXISTING`).

So we will use the following matrices as input:

- `BOM` is an $n \times n$ matrix, where n is the number of parts included in MRP calculation. This Matrix will represent the BOM just like the input-output matrix.

- `DEMAND` is an $1 \times n$ matrix, which represents the demand of the single items along the columns.

- `STOCK` is an $1 \times n$ matrix, which represents the current stock in the warehouse for the single items along the columns.

- `EXISTING` is an $1 \times n$ matrix, which represents the existing confirmed replenishment orders of the single items along the columns.

```
REQUEST =

     1     2     1     2     2     2
     0     1     0     1     1     0
     0     0     1     0     0     2
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1

REQUIREMENT =

   100   200   100   200   200   200
```

Figure 4.17.: Request Matrix and Requirement Matrix

- `LT` is an $1 \times n$ matrix, which represents the lead time for the single items along the columns.

The first step is to calculate the `REQUIREMENT` from `DEMAND` matrix. The mcode is as follows and the resulting matrices for our example is shown in figure 4.17 :

```
% Calculating Requirement
RMATRIX = eye(ncols)-BOM ;
REQUEST = inv(RMATRIX);

% Calculate the replenishment matrix
REQUIREMENT= DEMAND*REQUEST;
```

`REQUEST` is an $n \times n$ matrix, which represents the requirements for all components to produce one item. `REQUIREMENT` is an $n \times n$ matrix, which represents the requirement of all components to produce the demand. Till now the process is essentially the same.

The difference comes when we start considering the Lead Time as a matrix `LT` a $1 \times n$ matrix, which represents the lead time of the single items along the columns.

Lead time, as explained earlier, is the time required from the time (date in our case) in which an order is placed to the time in which it is delivered. In our case we assume that the date in which the order is placed to be the same as the date in which the demand first appears. In all current ERP systems the Lead times are considered a property of the item master data and constant for a single item during the requirements calculation. There are as many columns as there are items. Each value corresponds to the lead time of the corresponding

```
REQUIREMENT_LT =

     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     0     0     0
   100   200   100   200   200   200
```

Figure 4.18.: Requirement Matrix with Lead Times

component in the column of the BOM matrix. A positive lead time in reality means a delay. So we can treat the lead times as delays, implying that the replenishment order should be placed that much earlier in time. A lead time of 5 days means if you plan to use the item on day 5 you should order it on day 1. If you have the demand right on day one means the order should have been placed at least 5 days earlier.

For example the `LT` matrix `[1,5,3,2,2,1]` means that for `FG1` we have a lead time of 1 time bucket, for `C1` a lead time of 5 time buckets etc. A lead time of 5 time bucket for `C1` means that we have to place the order 5 days earlier than the day in which it is needed to produce FG1.

To include this delay, we will augment the `REQUIREMENT` matrix adding zeroes on the top, with the number of rows containing zeros equal to the maximum among the lead times, resulting in the `Requirement_LT` matrix as in figure 4.18. We add it on top since in this manner we can shift an entire column up according to the corresponding value of that column in the lead time matrix.

```
REQUIREMENT_LT=[zeros(max(LT),n);REQUIREMENT];
```

Now we should shift the columns, which represents single items or components, in the `Requirement_LT` matrix up or down according to the corresponding value in the `LT` matrix. Continuing with our example, we will shift the all values of the first column (which represents the $FG1$) one row up, the second column (which represents the component $C1$) 5 rows up etc to get the result shown in figure 4.19.

To shift the single columns up we will use sparse and full matrices [27][87][77] functions of Matlab [52].

- `S = sparse(A)` converts a matrix into sparse form by squeezing out any zero elements. If a matrix contains many zeros, converting the matrix to sparse storage saves a lot of memory.

- `S = sparse(m,n)` generates an $m \times n$ all zero sparse matrix.

```
REQUIREMENT_FINAL =

     0   200     0     0     0     0
     0     0     0     0     0     0
     0     0   100     0     0     0
     0     0     0   200   200     0
   100     0     0     0     0   200
     0     0     0     0     0     0
```

Figure 4.19.: Requirement Matrix with Lead Times Applied

- `S = sparse(i,j,v)` generates a sparse matrix S from the triplets i, j, and v such that $S(i(k), j(k)) = v(k)$. The $max(i) \times max(j)$ output matrix has space allotted for $length(v)$ non zero elements. sparse adds together elements in v that have duplicate subscripts in i and j. If the inputs i, j, and v are vectors or matrices, they must have the same number of elements. Alternatively, the argument v and/or one of the arguments i or j can be scalars.

- `S = sparse(i,j,v,m,n)` specifies the size of S as $m \times n$.

`A = full(S)` converts sparse matrix S to full storage organization, such that $issparse(A)$ returns logical 0 (false).

We will also use Fast Fourier Transform [11][94]. The Fourier transform and its inverse convert between data sampled in time and space and data sampled in frequency.[52]. The time-variant nature of this system also legitimates the use of Fast Fourier Transforms[38]. The resulting matrices as shown in the following algorithm provides us with an optimal data structure that has the advantages of both storage memory optimization as well as calculation optimization.

- `Y = fft(X)` Create a vector and compute its Fourier transform.

- `X = ifft(Y)` computes the inverse discrete Fourier transform of Y using a fast Fourier transform algorithm. X has the same size of Y. If Y is a matrix, then $ifft(Y)$ returns the inverse transform of each column of the matrix. `X = ifft(_,symflag)` specifies the symmetry of Y. For example, `ifft(Y,'symmetric')` treats Y as conjugate symmetric. For nearly conjugate symmetric vectors, you can compute the inverse Fourier transform faster by specifying the 'symmetric' option, which also ensures that the output is real. `.*` indicates element-wise multiplication

The mcode for what we just described is as follows:

```
[nrows,ncols] = size(REQUEST_LT);
```

```
STOCKPROGRESSION =

     0   500    50    50     0   500
     0   400    60    60     0   540
     0   500    70    70     0   580
     0   600   -20    80     0   620
     0   700   -10  -110  -200   660
  -100   800     0  -100  -200   500
  -100   900    10   -90  -200   540
```

Figure 4.20.: Stock Progression Matrix

```
TEMP = full(sparse(mod(-LT,nrows)+1,1:ncols,1,nrows,ncols));
REQUEST_FINAL = zeros(ncols);
REQUEST_FINAL = ifft(fft(REQUEST_LT).*fft(TEMP),'symmetric');
REQUEST_FINAL = round(REQUEST_FINAL);
```

At this point we can introduce the stock matrix `STOCK` and existing replenishment orders `EXISTING`. `STOCK` is an $1 \times n$ matrix, representing the current stock in the warehouse of the single items along the columns. `EXISTING` is a $1 \times n$ matrix as discussed earlier, which represents the existing confirmed replenishment orders of the single items along the columns. We can directly use a pure birth-death process [48] [9] which is a special case of continuous-time Markov process where the state transitions are of only two types: "births", which increase the state variable by one and "deaths", which decrease the state by one. A positive entry in the `EXISTING` matrix means a "birth" in our case and a positive entry in the `REQUEST_FINAL` matrix means a "death". We will subtract `REQUEST_FINAL` from `EXISTING` and then do a cumulative sum along the columns of this resulting matrix concatenated with the current stock matrix `STOCK` as shown below. The result is the progression of the stock expressed in the `STOCKPROGRESSION` matrix of the same size as that of `Requirement_LT` matrix, $n + max(LT) \times n$. The result of our example is shown in figure 4.20.

`B = cumsum(A)` returns the column wise cumulative sum of `A` starting at the beginning of the first array dimension in `A` whose size does not equal 1. If `A` is a matrix, then `cumsum(A)` returns a matrix containing the cumulative sums for each column of `A`. $C = cat(dim, A_1, A_2, \cdots, A_n)$ concatenates $A_1, A_2, \cdots, A_n$ along dimension `dim`.

```
STOCKPROGRESSION = cumsum(cat(1,STOCK,
                   (-1*REQUIREMENT_FINAL)+ EXISTING));
```

Finally we calculate the `MRP` matrix from this `STOCKPROGRESSION` matrix by removing zeros and multiplying it with -1 to create the `MRP` matrix as shown below. `MRP` matrix is of the same size as that of `Requirement_LT` matrix,

```
MRP =

     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0    20     0     0     0
     0     0    10   110   200     0
   100     0     0   100   200     0
   100     0     0    90   200     0
```

Figure 4.21.: MRP Matrix

$n + max(LT) \times n$. The final result of the example we are discussing is shown in figure 4.21.

```
MRP=STOCKPROGRESSION;
MRP(MRP>0)=0;
MRP = -1 * MRP;
```

The complete mcode for this extended matrix approach is shown below:

```
% Input data
LT = [1,5,3,2,2,1];
BOM = [0,2,1,0,0,0;0,0,0,1,1,0;0,0,0,0,0,2;
            0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];
DEMAND=[100,0,0,0,0,0];
STOCK =[0,500,50,50,0,500];
EXISTING =[0,100,10,10,0,40];

% Initializations
[nrows,ncols]=size(BOM);
REQUEST=zeros(nrows,ncols);
REQUIREMENT=zeros(ncols);

% Calculate Request
RMATRIX = eye(ncols)-BOM;
REQUEST = inv(RMATRIX);

% Calculate the replenishment matrix based on requirement matrix
% and demand matrix
REQUIREMENT= DEMAND*REQUEST;

% add space before to take into
% consideration the lead time
REQUIREMENT_LT=[zeros(max(LT),ncols);REQUIREMENT];
```

```
% Recalculate the replenishment matrix taking into consideration
% the demand matrix
[nrows,ncols]=size(REQUIREMENT_LT);
TEMP=full(sparse(mod(-LT,nrows)+1,1:
                              ncols,1,nrows,ncols));
REQUIREMENT_FINAL=zeros(ncols);
REQUIREMENT_FINAL = ifft(fft(REQUIREMENT_LT).*fft(TEMP),'symmetric');
REQUIREMENT_FINAL = round(REQUIREMENT_FINAL);

% Consider Replenishment orders
STOCKPROGRESSION = cumsum(cat(1,STOCK,
                   (-1*REQUIREMENT_FINAL)+ EXISTING));

% Calculate MRP
MRP=STOCKPROGRESSION;
MRP(MRP>0)=0;
MRP = -1 * MRP;
```

4.4.3. Expanded Requirements Calculation using Multidimensional Matrices

With the earlier extension we were able to include the lead time as well as stock in the MRP calculations. We still have to consider the possible changes in BOM with time. As mentioned earlier, there can be variations in BOM in the future both in the part list of any given item as well as the quantity of the parts used. This could be caused by a revision of the product structure, an improvement or simplification of the product or component, release of a new version of a component or obsolescence of an already existing one.

We will further extend the matrix method to evaluate the situation from a time-variant point of view. We see MRP as a system whose behaviour changes with time, so that the system will respond differently to the same input at different times. Moreover this system is not stationary. The second extension proposed is to overcome these shortcomings. The idea is to move on to a time varying dynamic system, adding another dimension to all the above matrices. This new dimension will represent the time horizon. This practically means a revision of all the matrices and a radical rethinking of how we view the various matrices (the data structure) and the algorithm itself.

Consider the `BOM` matrix as representing the BOM, where T is the planning horizon. This time varying `BOM` matrix will be of size $n \times n \times T$, n being the number of components. For each t, $1 \leq t \leq T$, BOM_t will be a $n \times n$ matrix. In this way if there is a change in an element of the BOM during a specific point in time t, from that point onwards that particular element value will be different. The change in quantity, increase, decrease or removal of the item from BOM will thus be reflected in the matrix. If there are additional components we just increase the dimension n of the matrix to the quantity $\bar{n}$ required, just ensuring that the upper triangular form is maintained in all the $\bar{n} \times \bar{n}$ matrices along the dimension T.

The demand matrix `DEMAND` also takes this new dimension thus becoming a matrix of size $1 \times n \times T$ with the demand of individual time buckets appearing as vectors in the new dimension.

Since we have a new dimension added we cannot directly apply the calculation done earlier to calculate the requirement matrix. We will use a mixed approach by creating the MRP table as in traditional MRP calculation approach but use matrices to simultaneously calculate the requirements for all components for any specific time bucket from $t = 0 \cdots T$. We will have the following matrices:

- *Bill Of Materials* - `BOM(t)`: This matrix represents the BOM for each time bucket along the dimension T. This can be thought of as T tables of $n \times n$ input-output matrices . It is an $n \times n \times T$ matrix.

- *Demand* - `DEMAND(t)`: Represents the demand for the n parts along the dimension T. Each single vector $1 \times n$ is the demand for the n items at time bucket t. It is an $1 \times n \times T$ matrix.

- *Gross Requirements* - `GR(t)`: Represents the gross requirements during the entire time horizon. Each row represents the gross requirements for the n items at any specifi time bucket t. It is an $T \times n$ matrix.

- *Scheduled Receipts* - `SR(t)`: Represents the whole scheduled receipts during the entire time horizon. Each row represents the scheduled receipts for the n items at time bucket t. It is a $T \times n$ matrix.

- *Projected On Hand* - `POH(t)`: Represents the projected on hand values for all items during the entire time horizon. Each row represents the projected on hand values for the n items at time bucket t. It is a $T \times n$ matrix.

- *Net Requirements* - `NR(t)`: Represents the net requirements values for all items during the entire time horizon. Each row represents the net requirements for the n items at time bucket t. It is a $T \times n$ matrix.

- *Planned Order Receipts* - `POR(t)`: Represents the planned order receipts for all items during the entire time horizon. Each row represents the planned order receipts values for the n items at time bucket t. It is a $T \times n$ matrix.

- *Planned Order Releases* - `POX(t)`: Represents the planned order releases for all items during the entire time horizon. Each row represents the planned order releases values for the n items at time bucket t. It is a $T \times n$ matrix.
- *Inventory On Hand* - `INV(t)` : Represents the inventory on hand for all items during the entire time horizon. Each row represents the inventory on hand values for the n items at time bucket t. It is a $T \times n$ matrix with $INV(1)$ representing the current stock.
- *Work In Process* - `WIP(t)`: Represents the work in process for all items during the entire time horizon. Each row represents the work in process values for the n items at time bucket t. It is a $T \times n$ matrix.
- *Minimum Lot Size* - `MLS(t)`: Represents the minimum lot size for all items during the entire time horizon. Each row represents the minimum lot size values for the n items at time bucket t. It is an $n \times n \times T$ matrix.
- *Lead Time* - `LT(t)` It is a $1 \times T$ matrix which represents the lead time of n items.

The algorithm that we will follow is summarised in figure 4.22.
We can divide this algorithm into the following five steps:

1. **Calculation of Request Matrix**: The first step is to calculate the Request matrix from the `BOM` and `DEMAND` matrices, as in the algorithm we saw earlier. We will move along the time dimension T and for each time bucket t calculate the input-output matrix for the respective `BOM` and `DEMAND` matrices at time t. The calculation is done for all n items.

```
REQUEST=zeros(n,n,t);
for nloop = 1:t
RMATRIX = eye(ncols)-BOM(:,:,nloop);
REQUEST(:,:,nloop) = inv(RMATRIX);
end
```

   The result is the `REQUEST` matrix of size $n \times n \times T$ just like the previous algorithm except that there are T matrices each representing the specific request at the respective time bucket t.

2. **Calculating the Gross Requirements**: The next step is to calculate the Gross Requirements. Unlike traditional MRP calculations where we calculate the gross requirement one by one for all items, we calculate the gross requirements of all the items at the same time. The biggest challenge here is to calculate correctly the gross requirements of the lower levels of BOM in a multi level BOM.

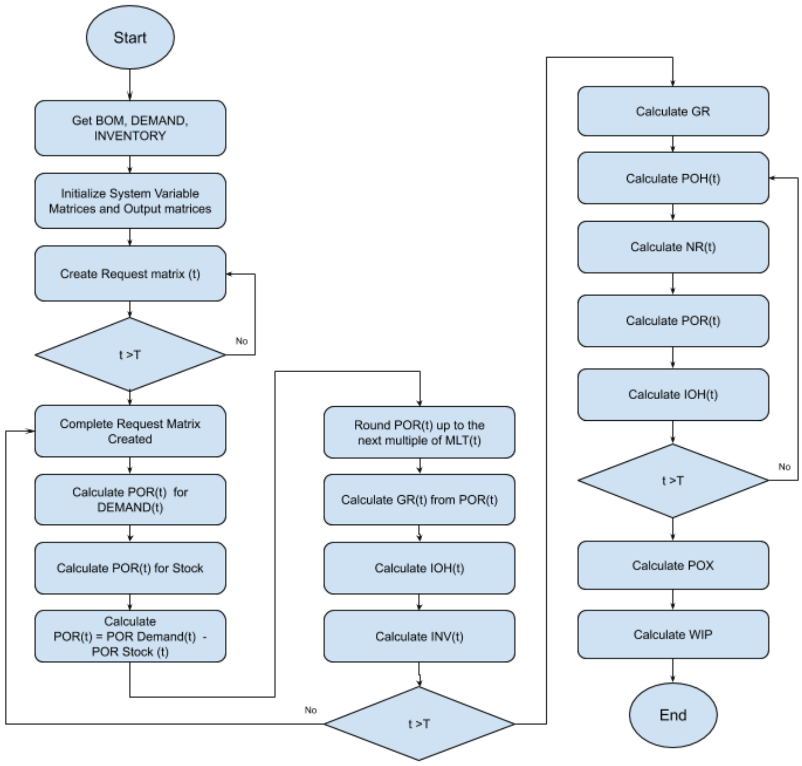

Figure 4.22.: MRP Matrix Flow

In the original input-output method the gross requirement is calculated without considering the stock, as if everything has to be produced or purchased. It is not enough to subtract the stock as it is from the Requirement Matrix seen earlier since the gross requirement of a lower level is affected by the stock of the upper levels along with the demand. Let us explain this with our example used in previous discussions. The BOM of this item is shown in the figure 4.3. The finished item $FG1$ is composed of two sub-assemblies $C1$ and $C2$, two pieces of $C1$ and one of $C2$. $C1$ is made from one piece each of $C3$ and $C4$ while $C2$ is made from two pieces of $C5$. So if we have a demand of 100 pieces of $FG1$ we can see that there is a derived demand of 200 pieces of $C1$ and 100 pieces of $C2$. This demand becomes the dependant demand for the next level and we need 200 pieces of $C3$, 200 pieces of $C4$ and 200 pieces of $C5$. This is the requirement that we will arrive at if we use the original method.

Let us suppose that we have a stock of 100 pieces of item $C2$ and 50 pieces of item $C5$. If we do a simple subtraction from the requirements we would see that we need no pieces of $C2$ and 150 pieces of $C5$. This is not completely true! The stock of 100 pieces of $C2$ is enough to cover the demand of 100 pieces and we need no additional pieces of $C2$. As a consequence we also do not need any pieces of $C5$ while the simple subtraction asked for 150 pieces. This is because the gross requirement was not propagated correctly to the lower levels of BOM.

What we need to subtract from the requirement is not the stock as it is but the stock subdivided into its components. We will consider the stock to be a "demand" to calculate the requirement of the components that would have been needed to produce that stock. We then subtract this requirement for stock from that for the demand. This we will do for all the individual matrices for every time bucket t.

What we essentially get at any time bucket t will be the requirement that we previously called Planned Order Requests (`POR`). So we calculate the `POR` of the inventory from that of the stock present in the inventory. We then remove negatives from this since negative value essentially means that we do not have any `POR` in that time bucket t.

Since we need to consider all the dependant demands, minimum lot sizes, inventory on hand and scheduled receipts, we go up doing a u-turn and calculate the gross requirement from the `POR` by simply multiplying it with the `BOM` matrix at that time bucket t. This is the same as subtracting the stock from demand and multiplying it with BOM and the Leontief inverse of the BOM. The improvement over the input-output method can be expressed by the equation 4.10 *This is one of the turning points of* *this method.*

$$(I_n - B)^{-1} * (D - S) \tag{4.10}$$

at each time bucket t Where D is the Demand, S the stock and B the BOM. I_n is an identity matrix with n the number of parts.

We now round up POR to the next higher multiple of the corresponding minimum lot size MLT just like we did for equation 3.12. After this we update the new stock level for the next time bucket $t + 1$ by

$$IOH(t, :) = IOH(t - 1, :) + SR(:, :, t) + POR(t, :) - GR(t, :).$$

At the end of each loop we update the Gross Requirement GR once again.

The mcode routine for this part is as follows:

```
for loop = 2:ndepth
 % Requirement from stock
 PORInventory=zeros(1,ncols);
 PORInventory = INV(:,:,loop-1)*REQUEST(:,:,loop);

 % Requirement from demand
 PORDemand=zeros(1,ncols);
 PORDemand = DEMAND(:,:,loop)*REQUEST(:,:,loop);

 % Requirement from demand - Requirement from stock
 PORDIFF = PORDemand - PORInventory;

 % Remove negatives
 PORDIFF(PORDIFF<0)=0;
 POR(loop,:) = PORDIFF;

 % Save Gross Requirements
 GR = POR * BOM(:,:,loop);

 % GR can also be calculated in an alternate way
 % GR(loop, :) = (DEMAND(:,:,loop)-INV(:,:,loop-1))
                  * BOM(:,:,loop)* REQUEST(:,:,loop);

 % Calculate Multiples
 TEMP3 = ceil(bsxfun(@rdivide,POR(loop, :),MLS(:, :, loop)));
 POR(loop, :) =  TEMP3 * diag(MLS(:, :, loop)) ;

 % Updating new stock
```

```
 IOH(loop, :) = IOH(loop-1, :) + SR(:, :, loop)
               + POR(loop, :) - GR(loop, :);
 INV(:,2:ncols,loop) = IOH(loop, 2:ncols);

 % Update Gross Requirements
 GR = POR * BOM(:,:,loop);
end
```

3. **Calculating the Net Requirements, Planned Order Receipts and Inventory on Hand**: They are calculated exactly like in classic MRP calculation, as shown in equations 4.2, 4.3 and 4.7. We do not take the *POR* calculated earlier into consideration and will recalculate it once again to include minimum lot sizes, current and progressive stocks also resulting from the planned receipts. The mcode is as follows:

```
% Calculate NR, POR and IOH
IOH(1, :) =INV(:,:,1);

for loop = 2:ndepth
  % Calculate POH
  POH(loop, :) = max(POH(loop-1, :) + SR(:, :, loop)
                                  - GR(loop, :) ,0);

  % Calculate NR
  TEMP = GR(loop, :) - SR(:, :, loop) - POH(loop-1, :);
  TEMP(TEMP<0)=0;
  NR(loop, :) = TEMP;

  % Calculate POR updated
  TEMPPOR= IOH(loop-1, :) + SR(:, :, loop) - GR(loop, :);
  TEMPPOR(TEMPPOR>0)=0;
  TEMPPOR(TEMPPOR<0)=1;
  POR(loop, :) = max(MLS(:, :, loop),(NR(loop, :)
                      - IOH(loop-1, :)))*diag(TEMPPOR);

  % Calculate IOH
  IOH(loop, :) = IOH(loop-1, :) + SR(:, :, loop)
                          + POR(loop, :) - GR(loop, :);
end

%Remove negatives from IOH
IOH(IOH<0)=0;
```

4. **Calculating the Planned Order Releases**: the POR is calculated exactly as defined earlier in equation 4.6, with the difference that this time we are dealing with matrices.

```
% Recalculate the replenishment matrix POX
TEMP = [zeros(max(LT),size(POR,2)); POR];
[nrows,ncols] = size(TEMP);
TEMP2 = full(sparse(mod(-LT,nrows)+1,1:ncols,1,nrows,ncols));
POX = round(ifft(fft(TEMP).*fft(TEMP2),'symmetric'));

% Sum all past demands in first time bucket
TEMP3 = cumsum(POX(1:max(LT)+1,:), 1);
POX(max(LT)+1,:) = TEMP3(max(LT)+1,:);
POX = POX(max(LT)+1:nrows,:);
```

5. **Calculating the Work In Process**: The WIP is calculated as in equation 4.8, once again using matrices.

```
% Calculate WIP
for loop = 2:ndepth
  WIP(loop, :) = WIP(loop-1, :) + POX(loop, :) -POR(loop, :);
end
WIP(WIP<0)=0;
```

The complete mcode for the extended multidimensional matrix approach to MRP is shown in appendix B.2.

For our example, the results with traditional MRP calculation is shown in figure 4.23 and the results using the extended matrix approach is shown in figure 4.24. We can see that they perfectly match each other.

4.4.4. Test results

One of the main challenges in dealing with the Material Requirements Planning problem was in finding a suitable data structure along with a suitable algorithm to deal efficiently with the very large number of data, in order to minimize the calculation resource overhead needed and the execution time necessary.

We have modelled the MRP as a dynamic system, whose defining matrices and inputs may vary with time, which evolves in time and is represented by three dimensional matrices that allows efficient computations.

This allows to apply further notions of systems engineering and system modelling. For instance, we can model the demand fluctuations as noise in this system.

GR

C1	C2	C3	C4	C5
0	0	0	0	0
200	100	0	0	200
200	100	0	0	200
200	100	150	150	200

POR

C1	C2	C3	C4	C5
0	0	0	0	0
0	100	0	0	0
0	100	0	0	0
150	100	100	150	100

SR

C1	C2	C3	C4	C5
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

POX

C1	C2	C3	C4	C5
0	100	0	0	0
0	100	0	0	0
150	100	100	150	100
150	100	150	150	200

POH

C1	C2	C3	C4	C5
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

IOH

C1	C2	C3	C4	C5
500	50	50	0	500
300	50	50	0	300
100	50	50	0	100
50	50	0	0	0

NR

C1	C2	C3	C4	C5
0	0	0	0	0
200	100	0	0	200
200	100	0	0	200
200	100	150	150	200

WIP

C1	C2	C3	C4	C5
0	0	0	0	0
0	0	0	0	0
150	0	100	150	100
150	0	150	150	200

Figure 4.23.: MRP Tables for all components

```
GR =

     0     0     0     0     0     0
     0   200   100     0     0   200
     0   200   100     0     0   200
     0   200   100   150   150   200
POH =

     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
SR  =

     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
NR =

     0     0     0     0     0     0
     0   200   100     0     0   200
     0   200   100     0     0   200
     0   200   100   150   150   200
```

```
POR =

     0     0     0     0     0     0
     0     0   100     0     0     0
     0     0   100     0     0     0
     0   150   100   100   150   100
POX =

     0     0   100     0     0     0
     0     0   100     0     0     0
     0   150   100   100   150   100
     0   150   100   150   150   200
IOH =

     0   500    50    50     0   500
     0   300    50    50     0   300
     0   100    50    50     0   100
     0    50    50     0     0     0
WIP =

     0     0     0     0     0     0
     0     0     0     0     0     0
     0   150     0   100   150   100
     0   150     0   150   150   200
```

Figure 4.24.: MRP Tables for all components in Matrix Form

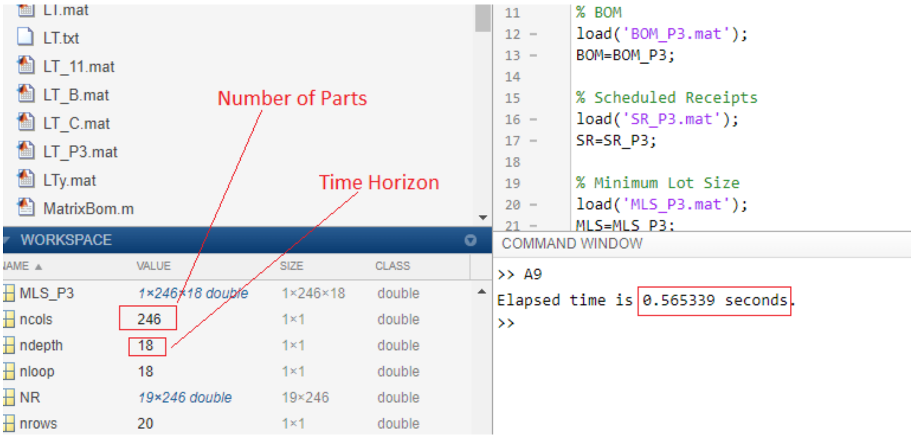

Figure 4.25.: MRP Calculation for 246 parts

We created a 1000 × 1000 × 700 matrix for the BOM in a test environment, simulating a 1000 part manufacturing facility in a two year time frame, considering a daily demand projection. The Demand Matrix was 1 × 700. The resulting DR_LT matrix had the size of 700 × 1000. The whole calculation took about 6 seconds in Matlab running on a Windows 10 a tablet with Intel Z8350 processor, RAM 2 GB. The same data set on a distributed AWS cloud running two traditional ERPs took little less than 10 minutes on the first and 9 minutes on the second. In the last part they do EOQ and MOQ calculations that we are not doing at the moment. But this took a couple of additional minutes on both systems that we subtracted from both the systems. We do not know what exact algorithms are used in these systems (SAP and BAAN) but from the log displayed it seems that they are spanning the BOM tree level by level distributing the demand to the various members. Obviously this is the only comparison test that we could do since all the proprietary ERP systems does not reveal the algorithms they use. In the future it would be nice to do a bench marking with the algorithms that they use.

We also did a real world test with in a small manufacturing facility with 246 valid part numbers in MRP. The elaboration took 0.57 seconds in Matlab online version, as shown in Figure 4.25. The BOM Matrix was of size 246 × 246 × 18 for the MRP horizon of 18 time buckets (Months in this specific case), with the results obtained from the standard ERP system used.

The results obtained are really promising and open the way to further research in this direction, may be implementing in some existing ERPs to field test the practical effectiveness of this approach.

4.5. Advantages

The new approach proposed have many significant advantages over the classical approach:

- Drastic reduction in calculation time: We can think of quick, constant updates of MRP and see the implications of any new information in the future helping us to act accordingly to arrive at a desired future output.
- Simple and straightforward data structure: There are just matrices which could be memorized straightforward without converting them to relational database constructs.
- Possibility of creating a sort of time machine: By memorizing the matrices in different time periods it becomes easy to consult the different scenarios at different periods of time. It also helps reverting back to past scenarios or jumping to future time periods quickly.
- Scenario analysis: Memory efficiency and elaboration speed of the proposed approach can help quick analysis of different scenarios with changing conditions like BOM, demand or stock variations. With classical approach this is already possible and all ERP systems implement some sort of scenario analysis The difference is in speed and less memory footprint required. We can also eliminate the need for a dedicated test environment where to do any scenario analysis we have to painstakingly copy all the data at any instance of time to have meaningful elaborations. Since at the end of the game we have matrices for all information, comparing and analysing matrices are way easier than analysing MRP tables where the human action is vital, making the analysis almost completely manual.
- Local and partial updates: We can easily evaluate the effects of small changes by doing only local and partial calculations on matrices without running the entire algorithm.
- Useful mathematical framework already available: we can apply all the concepts related to matrices and powerful matrix manipulating algorithms directly.

Just as in MPS, we calculate the MRP for the entire time horizon and how the requirements and stock progresses is calculated along the entire range. Following the MPC methodology only the first time buckets are taken into serious consideration. The exact number of time buckets considered, and the importance given to each one of them depends on the company policy. The future output can act as feed forward corrections to mantain the system output close to the desired output. The MPS matrix and MRP matrix can be considered as two sides of the same coin, giving complementary information that can complete the inventory planning along with stock considerations which we will discuss in the next chapter.

4.6. Conclusion

This chapter concludes the second part of inventory analysis with the presentation of a new multidimensional matrix approach to MRP following a multidimensional matrix approach to MPS. We can combine both of them to get a rapid requirement planning environment which gives an almost real time situation of the different possible scenarios without waiting for elaborate and time consuming calculations. This is an important improvement over all current ERP systems execute this as a batch job at the maximum once per day, mostly during inactive hours of the system. The method started in chapter 3 is completed in this chapter with the application to the material requirement planning calculations. In the coming chapters we will see the application of this modelling approach to to stockout forecasting (chapter 5), direct labour capacity calculations (chapter 6) and to forecast cash flow (chapter 7).

This main ideas of this chapter was also briefly presented in an article in International Conference on Control, Automation and Diagnosis (ICCAD) at Grenoble, France [69].

In our discussions we closely follow the framework of the classical approach has been maintained to help understanding the new approach. If we do not want to use the MRP table we can also think of alternative ways of representation, with a much simpler matrix with only the required and relevant information.

Another possibility is to couple the resulting matrices directly to EDI (Electronic data interchange) systems [26] [31] making it possible to do a distributed calculation of material requirement at both ends: at the supplier end as well as the customer end simultaneously and update just the individual values of any particular matrix. This brings a tremendous improvement over supply chain integration.

Chapter 5.

Inventory Planning: Inventory Management

When we speak of inventories in a supply chain context we are really speaking about the stocks. Stocks acts as a buffer between supply and demand [98] points in a supply chain. In an ideal world, intermediate and final stocks are not needed since goods, at all stages of production process, are produced only when there is a consumption: what enters exits.

A correct inventory management is essential to the successful operation of any company. First of all, stock is money stored in the warehouse. Money in the sense of not just the purchasing costs of the items but also the storage costs and missed investment opportunities. The money stored as stock could be better invested elsewhere. In lean manufacturing, inventory is considered as *Muda* Japanese term meaning waste used frequently in lean literature [97]. It is holds the second place among the 7 wastes defined first by Taiichi Ohno, father of the Toyota Production System: *transport, unnecessary inventories, unnecessary motions, waiting, overproduction, inappropriate processing and defects* [65].

At the Inventory planning phase, the goal is to find the correct inventory level that minimises the money blocked as stock. At the same time we need enough stock to run the production or sales smoothly without interruptions due to stockouts. Stockout refers to the event where the inventory on hand is exhausted and the demand or customer order remain unmet until the next replenishment. If an item is not available for a customer order then four possible effects can occur [86]:

1. Customer agrees to wait for the item;
2. Customer back orders the item;
3. Customer cancels the order;
4. Customer cancels the order, and is no longer a customer.

In all these cases there is an associated cost. The goal of every company is to avoid this at all cost. The first step in this process is to correctly forecast possible stockout. Only then an efficient preventive or corrective action can be put in place.

To correctly forecast the stockout we need to know the correct inventory levels needed. To know the correct inventory levels we should have a reliable demand planning, master production schedule and material requirements planning. Everything is interconnected and accuracy of one reflects on the accuracy of the others.

In this chapter we will discuss the modelling of inventory using matrices using the same logic we used for modelling MPS and MRP. We will then develop this to have a reliable inventory forecasting method to calculate also possible stockouts.

5.1. Modelling Inventory using Matrices

For this discussion, we will use a simple factory model consisting of a warehouse with the inventory of both purchased items as well as manufactured items from which the customer orders are shipped. The warehouse is replenished by purchased orders as well as manufacturing orders.

From classic DDS theory applied to supply chain [71] we have:

$$I_{t+1} = I_t + e_t + u_t \tag{5.1}$$

where I_{t+1} is the inventory at time $t+1$, e_t is the total incoming items at time t and u_t is the open items that will be shipped to customers at time t without considering the lead time θ and with just a single echelon. A detailed study could be found in the Phd thesis of Paris Pennesi on Adaptive Model Predictive Control in the Inventory Control Problem [67].

Le there be n items in the inventory of the factory and let T be the total number of time periods for which we calculate the stock progression. We can express the current stock as a row vector with n columns, each column representing the stock for a specific part at time $\bar{t}$. We have to fix the order of the parts at the beginning since this order has to be respected in all successive matrices: column 1 refers to item 1, column 2 refers to item 2 etc. WE obtain a $T \times n$ matrix W representing the stock evolution in the time interval $[t_0,\ t_0 + T]$. Each element of row $\bar{t}$ represents the stock of a particular item present in the inventory of this matrix represent a time bucket t from 0 to T in sequence. We will call this matrix *Inventory matrix.* At $t = 0$ this matrix will have the current stock of the parts in row 1 and all other rows null.

As in any inventory calculations, we will apply the birth-death process to this matrix. We will call the births `Input Matrix` $I(t)$ and the deaths `Output Matrix` $O(t)$. *Input matrix* I is a $T \times n$ matrix with T the total number of periods in the planning horizon and n number of items. It contains only positive entries or 0 and represents replenishment orders at time t. The *Output matrix* O is also a $T \times n$ matrix with T number of periods in the planning horizon and n number of items. It contains either positive entries or 0 and represents sales orders or outgoing stock at time t.

In practical terms I sums up all the incoming items to the stock and O sums up all the outgoing items from the stock. The input matrix can be considered as the sum of all production and purchase orders which on their own can be expressed as separate matrices. The inputs matrix in essence brings additional items to the stock. The outputs refers to the consumption of stock either by customer orders or by production orders that converts them to other items. At any instance t, the stock $W(t)$ is given by equation 5.2.

$$W_t = W_{t-1} + I_t - O_t \tag{5.2}$$

$I(t)$ and $O(t)$ refers to the information known on inputs and outputs at time t and all future calculations from $t \cdots T$ are incrementally calculated on this available information.

To simplify things we can think of a *Difference Matrix* X which subtracts $O(t)$ from $I(t)$ for each t. This *Difference matrix* X is a $T \times n$ matrix with T number of periods in planning horizon and n number of items. It contains positive entries, negative entries as well as 0. It is the matrix sum of Input and Output matrices and represents sum of transactions at time bucket t. Thus the equation 5.2 becomes simpler as in 5.3.

$$W_t = W_{t-1} + X_t \tag{5.3}$$

Each row of the matrix W at time t, the vector $W(t)$, gives the current situation of the stock at that specific time bucket resulting from the inputs, the outputs and the current stock.

If we have a row vector that gives the prices of the individual items, we can multiply this to each column. We can also create a diagonal matrix P of size $T \times n$ with the diagonals giving the price of the item in the corresponding column and multiply this directly with the Inventory Matrix W to get the *Inventory Value Matrix* V once again of dimension $T \times n$. So at any time bucket t the corresponding row of V gives the value of stock of the individual items in each columns. The sum of all the entries of this row gives the total stock value. Thus we have a rapid way of calculating the possible stock values for all time buckets from 0 to T.

An immediate use of this model is to calculate possible stockout values. If we consider only the items of V where the value is negative and set to zero all positive values and summing up the elements row-wise we have the total

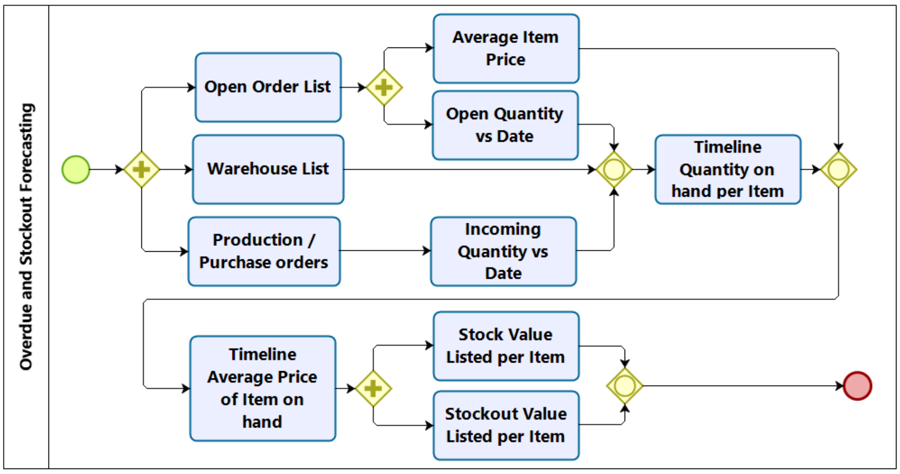

Figure 5.1.: Process Model

stockout value at all individual time buckets t.

5.2. Application of Matrix Approach

The Fig.5.1 shows the model of the process. The notation used is Business Process Model and Notation (BPMN) [23]. The process starts with 3 lists: Open orders list, warehouse list and production/purchase order list. For the number of columns we will use an ordered list of open items n and for the rows we will use the number of distinct time periods in the planning horizon. For example if we have a planning horizon of one month with daily subdivision, we will have 30 rows.

We can think of the inventory progression over the chosen horizon as a system as shown in Fig. 5.2 represented by the following matrices:

1. *Inventory Matrix* (W): a $T \times n$ matrix which represents the physical stock;

2. *Input Matrix* (I): a $T \times n$ matrix which represents replenishment orders;

3. *Output Matrix* (O): a $T \times n$ matrix which represents sales orders;

4. *Difference Matrix* (X): a n $T \times n$ matrix which represents the sum of input and output matrices;

5. *Price Vector* (P): a $1 \times n$ vector which represents the average sales price of each items;

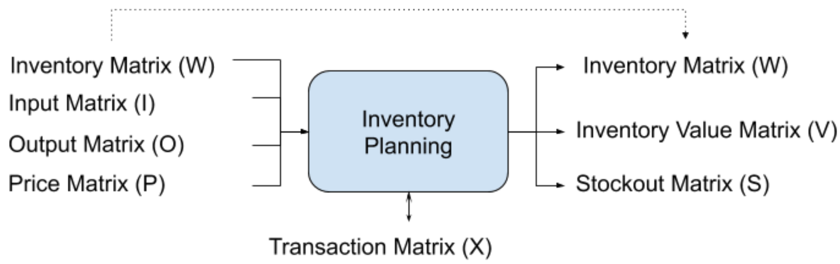

Figure 5.2.: Inventory Planning System View

6. *Inventory Value Matrix* (V): a $T \times n$ matrix which represents the inventory value for individual items in specific time buckets;

7. *Time Bucket Inventory Value Vector* (V_t): a $1 \times n$ matrix which represents the total inventory value per time bucket;

8. *Stockout matrix* (S): a $T \times n$ matrix which represents forecast and progression of stockout.

9. *Time Bucket Stockout Vector* (S_t): a $1 \times n$ matrix which represents the total stockout per time bucket.

The warehouse list directly becomes our Inventory Matrix W. The production and purchased order list sums up to form the Input Matrix and the Output Matrix created from the open orders list.

For our discussions, let us take the Input and Output matrices as shown in figures 5.3 and 5.4. As said earlier, the Input Matrix represents the replenishment orders and each entry i, j represents the incoming stock by production or purchase orders at time i for the jth item. Like all the previous matrices, the order of items must be fixed along the columns, same sequence of items for all matrices. The Output Matrix represents the outgoing stock mainly by sales orders. It can also be due to production orders that uses the finished goods as components for other finished goods.

Following equation 5.2, we have the Difference Matrix, which represents the sum of Input and Output matrices as shown in figure 5.5. This represents the sum of transactions that acts on the Inventory Matrix W.

At $t = 0$, we have the Inventory Matrix W as shown in figure 5.6. The first row is the current stock. After this, we sum the two matrices W and X as in equation 5.3. Any negative item in W indicates that there is a request that cannot be satisfied with current stock, which is what we mean by stockout.

```
I =

     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0   300    35     0     0     0     0    10     5
     0     0     0    20     0     0     0     0     0     0
     3     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     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0    55     0     0     0     0     0
     0     0     0     0     0   120     0    10     0    56
```

Figure 5.3.: Input Matrix

```
O =

     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0   100    24   100     0   360   780     0    50
     0     0     0     0     0     0     0    41     0    36
     0     0     0     0     5     5   100    12     0     0
     0     0     0     0     0     0     0     0     0     0
    20     0     0     5     5     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0    73     0     0
     0   240     0     0     0     0     0     0     0     0
```

Figure 5.4.: Output Matrix

```
X =

     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0   200    11  -100     0  -360  -780    10   -45
     0     0     0    20     0     0     0   -41     0   -36
     3     0     0     0    -5    -5  -100   -12     0     0
     0     0     0     0     0     0     0     0     0     0
   -20     0     0    -5    -5     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0    55     0     0   -73     0     0
     0  -240     0     0     0   120     0    10     0    56
```

Figure 5.5.: Difference Matrix

```
W =

     2    32     0     5    35    40    12   980    22    25
     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     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     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     0     0     0     0     0     0     0     0     0
```

Figure 5.6.: Inventory Matrix at $t = 0$

```
W =

     2     3     0     5    35    40    12   980    22    25
     2     3     0     5    35    40    12   980    22    25
     2     3   200    16   -65    40  -348   200    32   -20
     2     3   200    36   -65    40  -348   159    32   -56
     5     3   200    36   -70    35  -448   147    32   -56
     5     3   200    36   -70    35  -448   147    32   -56
   -15     3   200    31   -75    35  -448   147    32   -56
   -15     3   200    31   -75    35  -448   147    32   -56
   -15     3   200    31   -20    35  -448    74    32   -56
   -15  -237   200    31   -20   155  -448    84    32     0
```

Figure 5.7.: Inventory Matrix after $cumsum(W)$

The next step is to apply the cumulative sum over each columns. This step creates the inventory progression showing how the stock level progresses as t goes from 0 to T. We will overwrite the Inventory Matrix W with this result. W now will be as shown in figure 5.7. In each column we can see how the stock progresses.

From this updated Inventory Matrix W we can easily calculate the stock value V at any time bucket by removing the negative values from W and by row-wise multiplication with the Price Vector P. as shown in figure 5.8. It we retain just the negative values and then multiply W row-wise with the corresponding value of the vector P to get the stockout matrix S. We can do this by creating a diagonal matrix from P and multiply. To get the total inventory value and stockout of individual time buckets we do a cumulative sum along the rows and take the last column to get the two matrices VT representing the Time Bucket Inventory Value Matrix and ST representing Time Bucket Stockout Matrix

We can easily implement it as shown in the following lines of mcode:

```
% Transaction matrix
X = I - O;
% instantaneous inventory transactions
W = W + X;
% Inventory progression
W = cumsum(W,1);
% Inventory Value Matrix
V = W*diag(P);
V(V<0)=0;
```

```
P =

    105.07  84.23  327.28  21.52  18.98  52.34  56.63  197.76  250.47  139.64
```

Figure 5.8.: Price Matrix

```
V =

    210.14  2695.36         0  107.60  664.30  2093.60  679.56  193804.80  5510.34  3491.00
    210.14  2695.36         0  107.60  664.30  2093.60  679.56  193804.80  5510.34  3491.00
    210.14  2695.36  65456.00  344.32       0  2093.60       0   39552.00  8015.04        0
    210.14  2695.36  65456.00  774.72       0  2093.60       0   31443.84  8015.04        0
    525.35  2695.36  65456.00  774.72       0  1831.90       0   29070.72  8015.04        0
    525.35  2695.36  65456.00  774.72       0  1831.90       0   29070.72  8015.04        0
         0  2695.36  65456.00  667.12       0  1831.90       0   29070.72  8015.04        0
         0  2695.36  65456.00  667.12       0  1831.90       0   29070.72  8015.04        0
         0  2695.36  65456.00  667.12       0  1831.90       0   14634.24  8015.04        0
         0        0  65456.00  667.12       0  8112.70       0   16611.84  8015.04        0
```

Figure 5.9.: Inventory Value Matrix

```
% Calculate the stock value for single time buckets
VT = cumsum(V,2);
VT=VT(:,size(I,2));
% Retain only negative values in stockout
S = W*diag(P);
S(S>0)=0;
% Calculate stockout for single time buckets
ST = cumsum(S,2);
ST=ST(:,size(I,2));
```

So from the Price Matrix P as given in figure 5.8, applying the above algorithm we get the Inventory Value Matrix V as in figure 5.9, Time Bucket Inventory Value Matrix VT as in figure 5.10. The resulting Stockout matrix S is as in figure 5.11 and Time Bucket Stockout Matrix ST as in figure 5.12.

5.3. Use

- High inventory detection: We can see in Figure 5.13 the areas with high inventories (b) that are not used in a given time frame. We can also see where the inventory is well managed (f) and where it is poorly managed (e), fig 5.14
- High inventory on which item, for example as in Figure 5.13 (b), Figure 5.14 (a) and (d) along with any stock anomalies or leaps as in Figure 5.13 (c) - Figure 5.14 (c);

```
VT =

   209256.70
   209256.70
   118366.46
   110688.70
   108369.09
   108369.09
   107736.14
   107736.14
    93299.66
    98862.70
```

Figure 5.10.: Time Bucket Inventory Value Matrix

```
S =

          0           0   0   0        0   0         0   0   0        0
          0           0   0   0        0   0         0   0   0        0
          0           0   0   0 -1233.70   0 -19707.24   0   0 -2792.80
          0           0   0   0 -1233.70   0 -19707.24   0   0 -7819.84
          0           0   0   0 -1328.60   0 -25370.24   0   0 -7819.84
          0           0   0   0 -1328.60   0 -25370.24   0   0 -7819.84
   -1576.05           0   0   0 -1423.50   0 -25370.24   0   0 -7819.84
   -1576.05           0   0   0 -1423.50   0 -25370.24   0   0 -7819.84
   -1576.05           0   0   0  -379.60   0 -25370.24   0   0 -7819.84
   -1576.05   -17519.84   0   0  -379.60   0 -25370.24   0   0        0
```

Figure 5.11.: Stockout Matrix

```
ST =

           0
           0
   -23733.74
   -28760.78
   -34518.68
   -34518.68
   -36189.63
   -36189.63
   -35145.73
   -44845.73
```

Figure 5.12.: Time Bucket Stockout Matrix

W =

						(d)	(e)	(f)	
2	3	0	5	35	40	12	980	22	25
2	3	0	5	35	40	12	980	22	25
2	3	200	16	-65	40	-348	200	32	-20
2	3	200	36	-65	40	-348	159	32	-56
5	3	200	36	-70	35	-448	147	32	-56
5	3	200	36	-70	35	-448	147	32	-56
-15	3	200	31	-75	35	-448	147	32	-56
-15	3	200	31	-75	35	-448	147	32	-56
-15	3	200	31	-20	35	-448	74	32	-56
-15	-237	200	31	-20	155	-448	84	32	0
	(a)	(b)			(c)				

Figure 5.13.: Time Bucket Inventory Value Matrix Analysis

V =

210.14	2695.36	0	107.60	664.30	2093.60	679.56	193804.80	5510.34	3491.00
210.14	2695.36	0	107.60	664.30	2093.60	679.56	193804.80	5510.34	3491.00
210.14	2695.36	65456.00	344.32	0	2093.60	0	39552.00	8015.04	0
210.14	2695.36	65456.00	774.72	0	2093.60	0	31443.84	8015.04	0
525.35	2695.36	65456.00	774.72	0	1831.90	0	29070.72	8015.04	0
525.35	2695.36	65456.00	774.72	0	1831.90	0	29070.72	8015.04	0
0	2695.36	65456.00	667.12	0	1831.90	0	29070.72	8015.04	0
0	2695.36	65456.00	667.12	0	1831.90	0	29070.72	8015.04	0
0	2695.36	65456.00	667.12	0	1831.90	0	14634.24	8015.04	0
0	0	65456.00	667.12	0	8112.70	0	16611.84	8015.04	0
		(a)		(b)	(c)			(d)	

Figure 5.14.: Time Bucket Inventory Value Matrix Analysis

- Rapid calculation of the value of stock at any tome point t and also possible time frame of stockout impact as in Figure 5.14 (b);
- Help find optimal inventory values for the whole stock, a subset of stock or for any specific item;
- We can easily calculate the evolution of the inventory for any given item as in Figure 5.13 (e) and (f);
- W and V gives an immediate vision of the current inventory evolution for all items both in terms of quantity as well as value;
- We thus have a receding horizon, with an increase in reliability with respect to other possible analyzes;
- Analysis of correct inventory management, which items, where the soft point of the inventory management is, and simulate the impact of any corrective actions on suppliers and stock levels;
- Review effectiveness of current and possible inventory policies. Since with the matrices of the current state we can implement any inventory policies as additional matrices that acts upon this system and provide the results

5.4. Implementation

The concept was applied in a real world situation in a manufacturing factory which produces as well as resells products. The algorithm was applied to the last stage of outgoing warehouse.

The factory had an average declared response time of a minimum of 3 days on customer orders. The factory, thus, could deal with normal fluctuations on customer orders with lead time greater than 3 days. To check the validity of the predicted results we assumed that we were unaware of the lead time in which it was possible to do something and that all actions were assumed to be taken at the earliest possible time.

The model was applied as it is and the result is given in figure 5.15. We take a subset of 8 day period from the stockout matrix and progressively check back the predicted value with what the final stockout value was. On day 0 we have the real stockout and the prediction for days 1-8. On day 1 we have the real value of stockout for day 1 and a corrected prediction for days 2-8 and so on. On each day we calculate the percentage variation of the real value between the previously predictions as in figure 5.16. This in turn is graphically represented as in figure 5.17.

The results clearly show a well marked transition from low to high fluctuation around 3-4 days, inside which the numbers remain in a ± 5 % difference from

	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8
Day 0	-€ 15.540,96	-€ 18.978,67	-€ 18.978,67	-€ 18.978,67	-€ 31.433,30	-€ 30.790,29	-€ 26.986,80	-€ 27.146,46	-€ 14.818,72
Day 1		-€ 19.358,25	-€ 19.358,25	-€ 19.358,25	-€ 33.947,96	-€ 33.869,32	-€ 21.589,44	-€ 27.146,46	-€ 14.818,72
Day 2			-€ 18.583,92	-€ 19.358,25	-€ 33.947,96	-€ 33.869,32	-€ 21.589,44	-€ 17.645,20	-€ 13.665,82
Day 3				-€ 19.358,25	-€ 33.947,96	-€ 34.546,70	-€ 22.021,23	-€ 18.527,46	-€ 13.665,82
Day 4					-€ 33.947,96	-€ 34.546,70	-€ 22.021,23	-€ 18.712,73	-€ 13.529,16
Day 5						-€ 33.683,03	-€ 20.920,17	-€ 18.712,73	-€ 11.905,66
Day 6							-€ 20.459,92	-€ 18.712,73	-€ 11.905,66
Day 7								-€ 18.899,86	-€ 12.024,72
Day 8									-€ 12.024,72

Figure 5.15.: Waterfall Chart of Day 1 to 8 stockout forecasting

	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8
Forecast Day - 1		2%	4%	0%	0%	-2%	-2%	1%	0%
Forecast Day - 2			-2%	0%	0%	-2%	-7%	1%	1%
Forecast Day -3				2%	0%	-1%	-7%	1%	1%
Forecast Day - 4					0%	-1%	-5%	2%	-11%
Forecast Day - 5						9%	-5%	7%	-12%
Forecast Day - 6							-24%	-30%	-12%
Forecast Day - 7								-30%	-29%
Forecast Day - 8									-29%

Figure 5.16.: Waterfall Chart of Day 1 to 8 stockout forecasting variation in Percentage

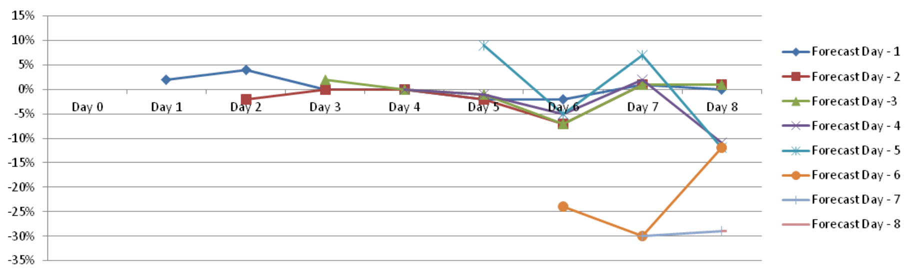

Figure 5.17.: Waterfall Chart of Day 1 to 8 stockout forecasting variation Graph

the predicted value. Even when there is oscillation, the resulting matrix gives a very precise indication as to where one should intervene and what is the magnitude of the result. Any corrective action can immediately be evaluated to determine its effect on overall results.

These matrices also provide an opportunity to do what-if analysis offline on the data set, may be using further optimization algorithms to find optimum actions. In this sample application the company did a quick Pareto with ABC-XYZ classification [82] on the resulting matrix to quickly sort out the worst performing items, which led to a quick identification of a poor performance from a supplier on a small subset of items.

Another application is to take snapshots of the state of the system with the inputs and outputs at any given time which can effectively act as a sort of time machine. It permits to go to a particular state at a particular time, to roll back to a past point before some given corrective action undertaken (or to a future point) and check its effectiveness, giving a very precious opportunity to identify indicators on effectiveness of corrective actions. These snapshots can be considered as a mathematical model of the factory system, opening the possibilities of applying a wide range of systems engineering considerations on the data set acquired.

Another analysis conducted was on checking the impact of a predicted material shortage from a particular supplier. The matrix helped determine the exact action points and the minimum supply required to avoid stockouts at month closings. This in turn helped avoid poor performance indicator results. Such an analysis was literally impossible with the current ERP of the company.

5.5. Conclusion

In this chapter we saw a practical application of modelling the inventory using the same approach of the previous chapters. In the practical implementation we saw that the application gave a reliable inventory forecasting method to calculate possible stockouts. In figure 5.17 we saw a successful application of this method in a real world business scenario.

Other than forecasting the stockout, this approach also has a variety of related applications extending the scope to offline what-if analysis, quick identification of a poor performance from a supplier on a small subset of items, create a sort of time machine. The mathematical model of the factory system developed also provides applications in a wide range of systems engineering.

This main ideas of this chapter was also briefly presented in an article in IEEE-2018 International Conference on Control Automation and Diagnosis, at Marrakech-Morocco [70].

Chapter 6.

Operations Planning

Operations planning provides a detailed insight into how the organizational goals are translated into reality, providing a plan for resource allocation. The demand planning which detailed the sales forecasted is used by inventory planning (MPS and MRP) to give a detailed plan on what is needed to satisfy the demand, when and in what quantity. Operations planning refers to the planning of other resources needed to satisfy the customer demand. Insufficient capacity may cause the entire production system to wildly and unpredictably fluctuate even with all input parameters are held constant.

Operations planning is a vast area in itself. In this chapter we will be focusing only on the capacity calculation of direct labour needed[84]. By direct labour we mean all the labour costs involved directly in the conversion of raw materials to finished goods. The main component of direct labour cost is the salary of the workers on production line. Indirect labour are all the other costs that the company sustains in its operation that cannot be directly associated to any stage of the production process. Office staff, maintenance costs, sales team etc falls into this category.

6.1. Capacity Requirements Planning

By Capacity Requirements Planning (CRP) we mean how the company calculates, plans and uses its facilities to satisfy the demand from the customer.

6.1.1. Basic Terms

- *Design Capacity*: Maximum capacity that the organization can offer to satisfy the demands of the customer in a given period.

- *Effective Capacity*: Real capacity that the organization can offer to satisfy the demand of the customers considering all known constraints like production and supply chain inefficiencies, quality problems etc.

6.1.2. Direct Labour Capacity Table

There is no commonly agreed capacity planning methodology and each ERP system defines its own one. The common factor among all the current ERP systems is the presence of some kind of tabular representation of how the capacity is planned and allocated for any given machine or production line. In this chapter we will not distinguish between production lines and machines. We will assume production line as a single entity without entering into further details of how it is composed. It will be our basic unit of production and from now on we will refer to it as *Production Unit*. For our study we will assume a generic capacity table with the following elements:

- *Working Days* `(N)`: Number of working days in the week at the production unit.
- *Working Shifts* `(F)`: Number of working shifts in a day at the production unit.
- *Working Hours* `(H)`: Number of working hours per shift at the production unit.
- *KOSU* `(K)`: Japanese word meaning "Manual Time"[39], is a key indicator in production control. It defines the level of productivity by identifying the true production time needed and so is a measure of productivity. It refers to the number of man hours it takes to produce one unit of a product, or to complete a process [45]. It is calculated by multiplying the number of workers involved in making a product by the number of hours used. That gives the total man hours. That number is then divided by the total number of units produced during that time. More on this in appendix A.8.
- *Line Scrap* `(U)`: Refers to the non-compliant parts which do not meet customer requirements in a production lot.
- *Line Inefficiency* `(V)`: Time lost due to inefficiencies in production like machine down times, inefficient production process etc.
- *Line Absenteeism* `(W)`: Time lost due to workers not presenting at their work.
- *Demand* `(D)`: Total number of products requested by the customer, the customer demand.
- *Pieces Per Day* `(P)`: Number of pieces to be produced per day according to production constraints to meet the customer demand. It is also refereed to as *Takt Time* [57],the rate of production needed to match the demand.

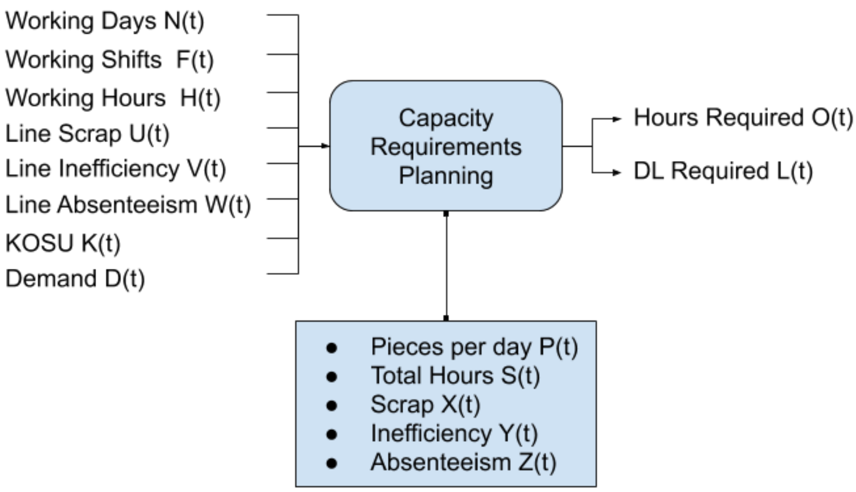

Figure 6.1.: CRP as a System

- *Total Hours* `(S)`: Total production hours needed to satisfy the demand from the customer.

- *Scrap* `(X)`: Total hours lost due to production of non compliant parts.

- *Inefficiency* `(Y)`: Total hours lost due to inefficiencies.

- *Absenteeism* `(Z)`: Total hours lost due to absenteeism.

- *Total Hours* `(O)`: Net hours needed for production considering all the inefficiencies. This is the main output of the system.

- *DL Required* `(L)`: Hours of direct labour needed, derived from net hours. As said earlier by Direct Labour we mean the labour involved in production on the production unit rather than administration, maintenance, and other support services.

Just as we did with MPS and MRP, we can approach the CRP as a system with the first 8 items of the Capacity Table as the input variables, the last two as the output variables and the rest as system variables as shown in figure 6.1.

From the production management we should have the input variables, namely Working Days $N(t)$, Working Shifts $F(t)$, Working Hours $H(t)$, Line Scrap $U(t)$, Line Inefficiency $V(t)$, Line Absenteeism $W(t)$ and KOSU $K(t)$. The

MRP provides with the Demand $D(t)$. From these inputs we calculate the system variables as follows:

$$Pieces\,per\,day = \frac{Demand}{Working\ Days} \tag{6.1}$$

$$Total\ Hours = \frac{Demand * KOSU}{3600} \tag{6.2}$$

$$Scrap = Total\ Hours * Line\ Scrap \tag{6.3}$$

$$Inefficiency = (Total\ Hours + Scrap) * Line\ Inefficiency \tag{6.4}$$

$$Absenteeism = \frac{Line\ Absenteeism}{1 - Line\ Absenteeism} * (Total\ Hours + Scrap) \tag{6.5}$$

$$Hours\ Required = Total\ Hours + Scrap + Inefficiency + Absenteeism \tag{6.6}$$

$$DL\ Required = \frac{\frac{Total\ Hours}{Working\ Hours * Working\ Shifts}}{Working\ Days} \tag{6.7}$$

Which can be expressed using variables evolving over time as in equations 6.8, 6.9, 6.10, 6.11, 6.12, 6.13 and 6.14:

$$P(t) = \frac{D(t)}{N(t)} \tag{6.8}$$

$$S(t) = \frac{D(t) * K(t)}{3600} \tag{6.9}$$

$$X(t) = S(t) * U(t) \tag{6.10}$$

$$Y(t) = (S(t) + X(t)) * V(t) \tag{6.11}$$

$$Z(t) = \frac{W(t)}{1 - W(t)} * (S(t) + X(t)) \tag{6.12}$$

$$O(t) = S(t) + X(t) + Y(t) + Z(t) \tag{6.13}$$

	M+0	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9	M+10
Working Days (N)	20	22	21	23	7	21	23	21	15	21	20
Working Shifts (F)	1	1	1	1	1	1	1	1	1	1	1
Working Hours (H)	8	8	8	8	8	8	8	8	8	8	8
Line Scrap (U)	0	0	0	0	0	0	0	0	0	0	0
Line Inefficiency (V)	0	0	0	0	0	0	0	0	0	0	0
Line Absenteeism (W)	-	-	-	-	-	-	-	-	-	-	-
KOSU (K)	750	750	750	750	750	750	750	750	750	750	750
Demand (D)	7.330	7.277	7.273	7.251	7.202	7.322	7.263	7.317	7.253	6.853	6.853
Pieces per day (P)	367	331	346	315	1.029	349	316	348	484	326	343
Total Hours (S)	1.527	1.516	1.515	1.511	1.500	1.525	1.513	1.524	1.511	1.428	1.428
Scrap (X)	15	15	15	15	15	15	15	15	15	14	14
Inefficiency (Y)	278	276	275	275	273	277	275	277	275	260	260
Absenteeism (Z)	-	-	-	-	-	-	-	-	-	-	-
Hours Required (O)	1.820	1.807	1.806	1.800	1.788	1.818	1.803	1.817	1.801	1.702	1.702
DL Required (L)	12	11	11	10	34	12	10	12	16	11	11

Figure 6.2.: MPS Matrix: Demand Forecast

	M+0	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9	M+10
Working Days (N)	20	22	21	23	7	21	23	21	15	21	20
Working Shifts (F)	1	1	1	1	1	1	1	1	1	1	1
Working Hours (H)	8	8	8	8	8	8	8	8	8	8	8
Line Scrap (U)	0	0	0	0	0	0	0	0	0	0	0
Line Inefficiency (V)	0	0	0	0	0	0	0	0	0	0	0
Line Absenteeism (W)	-	-	-	-	-	-	-	-	-	-	-
KOSU (K)	250	250	250	250	250	250	250	250	250	250	250
Demand (D)	706	632	627	595	527	696	613	688	598	606	606
Pieces per day (P)	35	29	30	26	75	33	27	33	40	29	30
Total Hours (S)	49	44	44	41	37	48	43	48	42	42	42
Scrap (X)	0	0	0	0	0	0	0	0	0	0	0
Inefficiency (Y)	7	7	7	6	6	7	6	7	6	6	6
Absenteeism (Z)	-	-	-	-	-	-	-	-	-	-	-
Total Hours (O)	57	51	51	48	43	56	49	55	48	49	49
DL Required (L)	0	0	0	0	1	0	0	0	0	0	0

Figure 6.3.: MPS Matrix: Demand Forecast

$$L(t) = \frac{\frac{O(t)}{H(t) * F(t)}}{N(t)} \tag{6.14}$$

The resulting DL CRP table for the 5 production units of our examples are as shown in figures 6.2, 6.3, 6.4, 6.5 and 6.6.

6.1.3. Matrix Representation of DL Capacity Table

Just as we did with MPS and MRP we can think of representing the capacity table using a matrix. The idea is to use 15 matrices each representing the variable which could be input, output or system variable. Each of them will be of size $1 \times T$ where T is our capacity planning horizon. They are as follows:

1. Working Days Matrix (N);

	M+0	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9	M+10
Working Days (N)	20	22	21	23	7	21	23	21	15	21	20
Working Shifts (F)	1	1	1	1	1	1	1	1	1	1	1
Working Hours (H)	8	8	8	8	8	8	8	8	8	8	8
Line Scrap (U)	0	0	0	0	0	0	0	0	0	0	0
Line Inefficiency (V)	0	0	0	0	0	0	0	0	0	0	0
Line Absenteeism (W)	-	-	-	-	-	-	-	-	-	-	-
KOSU (K)	250	250	250	250	250	250	250	250	250	250	250
Demand (D)	1.748	1.733	1.732	1.726	1.712	1.746	1.729	1.744	1.726	1.108	1.108
Pieces per day (P)	87	79	82	75	245	83	75	83	115	53	55
Total Hours (S)	121	120	120	120	119	121	120	121	120	77	77
Scrap (X)	2	2	2	2	2	2	2	2	2	1	1
Inefficiency (Y)	31	31	31	30	30	31	30	31	30	20	20
Absenteeism (Z)	-	-	-	-	-	-	-	-	-	-	-
Total Hours (O)	154	153	153	152	151	154	152	154	152	98	98
DL Required (L)	1	1	1	1	3	1	1	1	1	1	1

Figure 6.4.: MPS Matrix: Demand Forecast

	M+0	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9	M+10
Working Days (N)	20	22	21	23	7	21	23	21	15	21	20
Working Shifts (F)	1	1	1	1	1	1	1	1	1	1	1
Working Hours (H)	8	8	8	8	8	8	8	8	8	8	8
Line Scrap (U)	0	0	0	0	0	0	0	0	0	0	0
Line Inefficiency (V)	0	0	0	0	0	0	0	0	0	0	0
Line Absenteeism (W)	-	-	-	-	-	-	-	-	-	-	-
KOSU (K)	250	250	250	250	250	250	250	250	250	250	250
Demand (D)	2.315	2.297	2.296	2.288	2.272	2.312	2.292	2.311	2.289	1.649	1.649
Pieces per day (P)	116	104	109	99	325	110	100	110	153	79	82
Total Hours (S)	161	160	159	159	158	161	159	160	159	115	115
Scrap (X)	2	2	2	2	2	2	2	2	2	2	2
Inefficiency (Y)	41	40	40	40	40	41	40	41	40	29	29
Absenteeism (Z)	-	-	-	-	-	-	-	-	-	-	-
Total Hours (O)	204	202	202	202	200	204	202	204	202	145	145
DL Required (L)	1	1	1	1	4	1	1	1	2	1	1

Figure 6.5.: MPS Matrix: Demand Forecast

	M+0	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9	M+10
Working Days (N)	20	22	21	23	7	21	23	21	15	21	20
Working Shifts (F)	1	1	1	1	1	1	1	1	1	1	1
Working Hours (H)	8	8	8	8	8	8	8	8	8	8	8
Line Scrap (U)	0	0	0	0	0	0	0	0	0	0	0
Line Inefficiency (V)	0	0	0	0	0	0	0	0	0	0	0
Line Absenteeism (W)	-	-	-	-	-	-	-	-	-	-	-
KOSU (K)	800	800	800	800	800	800	800	800	800	800	800
Demand (D)	1.704	1.687	1.686	1.679	1.664	1.702	1.683	1.700	1.680	2.009	2.009
Pieces per day (P)	85	77	80	73	238	81	73	81	112	96	100
Total Hours (S)	379	375	375	373	370	378	374	378	373	446	446
Scrap (X)	8	7	7	7	7	8	7	8	7	9	9
Inefficiency (Y)	70	69	69	69	68	69	69	69	69	82	82
Absenteeism (Z)	-	-	-	-	-	-	-	-	-	-	-
Total Hours (O)	456	451	451	449	445	455	450	455	449	537	537
DL Required (L)	3	3	3	3	8	3	3	3	4	3	4

Figure 6.6.: MPS Matrix: Demand Forecast

2. Working Shifts Matrix (F);
3. Working Hours Matrix (H);
4. Line Scrap Matrix (U);
5. Line Inefficiency Matrix (V);
6. Line Absenteeism Matrix (W);
7. KOSU Matrix (K);
8. Demand Matrix (D);
9. Pieces Per Day Matrix (P);
10. Total Hours Matrix (S);
11. Scrap Matrix (X);
12. Inefficiency Matrix (Y);
13. Absenteeism Matrix (Z);
14. Hours Required Matrix (O);
15. DL Required Matrix (L).

We will apply all the system equations 6.8, 6.9, 6.10, 6.11, 6.12, 6.13 and 6.14 to these matrices to fill up the output matrices O of the hours required and L the direct labour required to satisfy the demand. The mcode is as follows with $.*$ and ./representing element-wise multiplication and division respectively:

```
% Input

% Working Days
N=[20,22,21,23,7,21,23,21,15,21,20];
% Working Shifts
F=[1,1,1,1,1,1,1,1,1,1,1];
% Working Hours
H=[7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5];
% Line Scrap
U=[0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01];
% Line Inefficiency
V=[0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18];
% Line Absenteeism
W=[0,0,0,0,0,0,0,0,0,0,0];
% KOSU
K=[750,750,750,750,750,750,750,750,750,750,750];
```

```
C =

    20    22    21    23     7    21    23    21    15    21    20
     1     1     1     1     1     1     1     1     1     1     1
   7,5   7,5   7,5   7,5   7,5   7,5   7,5   7,5   7,5   7,5   7,5
  0,01  0,01  0,01  0,01  0,01  0,01  0,01  0,01  0,01  0,01  0,01
  0,18  0,18  0,18  0,18  0,18  0,18  0,18  0,18  0,18  0,18  0,18
     0     0     0     0     0     0     0     0     0     0     0
   750   750   750   750   750   750   750   750   750   750   750
  7330  7277  7273  7251  7202  7322  7263  7317  7253  6853  6853
     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     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     0
     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0
```

Figure 6.7.: Input Matrix for Production Unit 1

```
% Demand
D=[7330,7277,7273,7251,7202,7322,7263,7317,7253,6853,6853];

% Output
P = D./N;                   % Pieces per Day
S = (D.*K)./3600;           % Total Hours
X = S.*U;                   % Scrap
Y = (S+X).*V;               % Inefficiency
Z = (W./(1-W)).*(S+X);      % Absenteeism
O = S+X+Y+Z;                % Hours Required
L = (O./(H.*F))./N;         % DL Required
```

The above algorithm means we have to do all the calculations one by one for all the single entries. So it is natural that we do the same reasoning done in earlier chapters to group them together into a single matrix. We will use a 15 row matrix with the first 8 rows (input variables) filled up with input data and the rest with zeros. In our example, for the Production Unit 1 this will be as shown in figure 6.7. Each row of this matrix was an independent matrix till now.

We will work exclusively on this matrix and apply all the system equations 6.8, 6.9, 6.10, 6.11, 6.12, 6.13 and 6.14 directly to this single matrix. The mcode is as follows:

```
% CRP Matrix for Production Unit 1
C(1)=[20,22,21,23,7,21,23,21,15,21,20];
C(2)=[1,1,1,1,1,1,1,1,1,1,1];
C(3)=[7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5];
```

C =

20	22	21	23	7	21	23	21	15	21	20
1	1	1	1	1	1	1	1	1	1	1
7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5
0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18
0	0	0	0	0	0	0	0	0	0	0
750	750	750	750	750	750	750	750	750	750	750
7330	7277	7273	7251	7202	7322	7263	7317	7253	6853	6853
366,5	330,77	346,33	315,26	1028,86	348,67	315,78	348,43	483,53	326,33	342,65
1527,08	1516,04	1515,21	1510,63	1500,42	1525,42	1513,13	1524,38	1511,04	1427,71	1427,71
15,27	15,16	15,15	15,11	15	15,25	15,13	15,24	15,11	14,28	14,28
277,62	275,62	275,46	274,63	272,78	277,32	275,09	277,13	274,71	259,56	259,56
0	0	0	0	0	0	0	0	0	0	0
1819,98	1806,82	1805,83	1800,36	1788,2	1817,99	1803,34	1816,75	1800,86	1701,54	1701,54
12,13	10,95	11,47	10,44	34,06	11,54	10,45	11,53	16,01	10,8	11,34

Figure 6.8.: CRP Matrix for Production Unit 1

```
C(4)=[0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01];
C(5)=[0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18];
C(6)=[0,0,0,0,0,0,0,0,0,0,0];
C(7)=[750,750,750,750,750,750,750,750,750,750,750];
C(8)=[7330,7277,7273,7251,7202,7322,7263,7317,7253,6853,6853];
C(9)=[0,0,0,0,0,0,0,0,0,0,0];
C(10)=[0,0,0,0,0,0,0,0,0,0,0];
C(11)=[0,0,0,0,0,0,0,0,0,0,0];
C(12)=[0,0,0,0,0,0,0,0,0,0,0];
C(13)=[0,0,0,0,0,0,0,0,0,0,0];
C(14)=[0,0,0,0,0,0,0,0,0,0,0];
C(15)=[0,0,0,0,0,0,0,0,0,0,0];

% Calculations for Production Unit 1
C(9)= C(8) ./ C(1);                      % Pieces per Day
C(10) = (C(8).*C(7))./3600;              % Total Hours
C(11) = C(10).*C(4);                     % Scrap
C(12) = (C(10)+C(11)).*C(5);             % Inefficiency
C(13) = (C(6)./(1-C(6))).*(C(10)+C(11));% Absenteeism
C(14) = C(10)+C(11)+C(12)+C(13);         % Hours Required
C(15) = (C(14)./(C(3).*C(2)))./C(1);     % DL Required
```

The resulting complete DL CRP Matrix for Production Unit 1 is shown in figure 6.8.

6.2. Multidimensional DL CRP Matrix

Once again we have the problem of calculating one by the DL CRP tables for all the single production units and then take the sum of the requirements. The proposal here is add a third dimension as we did in our previous discussions, with the third dimension representing the different production units. So the CRP matrix will be $15 \times T \times n$ there T is the time horizon for which the CRP calculation is done for n production units. n_i is the i^{th} production unit. So the $C(:,:,1)$ is the CRP matrix for the production unit 1, $C(:,:,2)$ is the CRP matrix for the production unit 2 and $C(:,:,n)$ is the CRP matrix for the production unit n. We can update our previous algorithm as shown in the following mcode:

```
[nrows,ncols,ndepth]=size(C);

% Output
for loop = 1:ndepth
   % Pieces per Day
   C(9,:,loop)= C(8,:,loop) ./ C(1,:,loop);
   % Total Hours
   C(10,:,loop) = (C(8,:,loop).*C(7,:,loop))./3600;
   % Scrap
   C(11,:,loop) = C(10,:,loop).*C(4,:,loop);
   % Inefficiency
   C(12,:,loop) = (C(10,:,loop)+C(11,:,loop)).*C(5,:,loop);
   % Absenteeism
   C(13,:,loop) = (C(6,:,loop)./(1-C(6,:,loop)))
                    .*(C(10,:,loop)+C(11,:,loop));
   % Hours Required
   C(14,:,loop) = C(10,:,loop)+C(11,:,loop)
                    + C(12,:,loop)+C(13,:,loop);
   % DL Required
   C(15,:,loop) = (C(14,:,loop)./(C(3,:,loop)
                    .*C(2,:,loop)))./C(1,:,loop);
end

% Total Direct Labor required
DL = sum(C(nrows,:,:),3);
```

We sum up all the individual direct labour requirements of the individual matrices to calculate the total direct labour requirements of all production lines in the 15th dimension of this matrix.

An executable version of this code is in appendix B.3. and the resulting matrices are as in figure 6.9 to 6.13.

C(:,:,1) =

20	22	21	23	7	21	23	21	15	21	20
1	1	1	1	1	1	1	1	1	1	1
7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5
0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18
0	0	0	0	0	0	0	0	0	0	0
750	750	750	750	750	750	750	750	750	750	750
7330	7277	7273	7251	7202	7322	7263	7317	7253	6853	6853
366,5	330,77	346,33	315,26	1028,86	348,67	315,78	348,43	483,53	326,33	342,65
1527,08	1516,04	1515,21	1510,63	1500,42	1525,42	1513,13	1524,38	1511,04	1427,71	1427,71
15,27	15,16	15,15	15,11	15	15,25	15,13	15,24	15,11	14,28	14,28
277,62	275,62	275,46	274,63	272,78	277,32	275,09	277,13	274,71	259,56	259,56
0	0	0	0	0	0	0	0	0	0	0
1819,98	1806,82	1805,83	1800,36	1788,2	1817,99	1803,34	1816,75	1800,86	1701,54	1701,54
12,13	10,95	11,47	10,44	34,06	11,54	10,45	11,53	16,01	10,8	11,34

Figure 6.9.: CRP Matrix: Production Unit 1

C(:,:,2) =

20	22	21	23	7	21	23	21	15	21	20
1	1	1	1	1	1	1	1	1	1	1
7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5
0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
0	0	0	0	0	0	0	0	0	0	0
250	250	250	250	250	250	250	250	250	250	250
706	632	627	595	527	696	613	688	598	606	606
35,3	28,73	29,86	25,87	75,29	33,14	26,65	32,76	39,87	28,86	30,3
49,03	43,89	43,54	41,32	36,6	48,33	42,57	47,78	41,53	42,08	42,08
0,49	0,44	0,44	0,41	0,37	0,48	0,43	0,48	0,42	0,42	0,42
7,43	6,65	6,6	6,26	5,54	7,32	6,45	7,24	6,29	6,38	6,38
0	0	0	0	0	0	0	0	0	0	0
56,95	50,98	50,57	47,99	42,51	56,14	49,44	55,49	48,23	48,88	48,88
0,38	0,31	0,32	0,28	0,81	0,36	0,29	0,35	0,43	0,31	0,33

Figure 6.10.: CRP Matrix: Production Unit 2

C(:,:,3) =

20	22	21	23	7	21	23	21	15	21	20
1	1	1	1	1	1	1	1	1	1	1
7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5
0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02
0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25
0	0	0	0	0	0	0	0	0	0	0
250	250	250	250	250	250	250	250	250	250	250
1748	1733	1732	1726	1712	1746	1729	1744	1726	1108	1108
87,4	78,77	82,48	75,04	244,57	83,14	75,17	83,05	115,07	52,76	55,4
121,39	120,35	120,28	119,86	118,89	121,25	120,07	121,11	119,86	76,94	76,94
1,82	1,81	1,8	1,8	1,78	1,82	1,8	1,82	1,8	1,15	1,15
30,8	30,54	30,52	30,41	30,17	30,77	30,47	30,73	30,41	19,52	19,52
0	0	0	0	0	0	0	0	0	0	0
154,01	152,69	152,6	152,07	150,84	153,84	152,34	153,66	152,07	97,62	97,62
1,03	0,93	0,97	0,88	2,87	0,98	0,88	0,98	1,35	0,62	0,65

Figure 6.11.: CRP Matrix: Production Unit 3

C(:,:,4) =

20	22	21	23	7	21	23	21	15	21	20
1	1	1	1	1	1	1	1	1	1	1
7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5
0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02
0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25
0	0	0	0	0	0	0	0	0	0	0
250	250	250	250	250	250	250	250	250	250	250
2315	2297	2296	2288	2272	2312	2292	2311	2289	1649	1649
115,75	104,41	109,33	99,48	324,57	110,1	99,65	110,05	152,6	78,52	82,45
160,76	159,51	159,44	158,89	157,78	160,56	159,17	160,49	158,96	114,51	114,51
2,41	2,39	2,39	2,38	2,37	2,41	2,39	2,41	2,38	1,72	1,72
40,79	40,48	40,46	40,32	40,04	40,74	40,39	40,72	40,34	29,06	29,06
0	0	0	0	0	0	0	0	0	0	0
203,97	202,38	202,3	201,59	200,18	203,7	201,94	203,62	201,68	145,29	145,29
1,36	1,23	1,28	1,17	3,81	1,29	1,17	1,29	1,79	0,92	0,97

Figure 6.12.: CRP Matrix: Production Unit 4

C(:,:,5) =

20	22	21	23	7	21	23	21	15	21	20
1	1	1	1	1	1	1	1	1	1	1
7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5	7,5
0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02
0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18	0,18
0	0	0	0	0	0	0	0	0	0	0
800	800	800	800	800	800	800	800	800	800	800
1704	1687	1686	1679	1664	1702	1683	1700	1680	2009	2009
85,2	76,68	80,29	73	237,71	81,05	73,17	80,95	112	95,67	100,45
378,67	374,89	374,67	373,11	369,78	378,22	374	377,78	373,33	446,44	446,44
7,57	7,5	7,49	7,46	7,4	7,56	7,48	7,56	7,47	8,93	8,93
69,52	68,83	68,79	68,5	67,89	69,44	68,67	69,36	68,54	81,97	81,97
0	0	0	0	0	0	0	0	0	0	0
455,76	451,22	450,95	449,08	445,06	455,23	450,15	454,69	449,34	537,34	537,34
3,04	2,73	2,86	2,6	8,48	2,89	2,61	2,89	3,99	3,41	3,58

Figure 6.13.: CRP Matrix: Production Unit 5

DL =

17,94	16,15	16,9	15,37	50,03	17,06	15,4	17,04	23,57	16,06	16,87

Figure 6.14.: CRP Matrix: Direct Labor Requirements

6.3. Conclusion

In this chapter we discussed how the multidimensional matrix approach developed to better calculate MPS and MRP can be applied also to direct labour capacity calculations. We concentrated on the calculation of the requirements for direct labour for a set of production units. The results were up to our expectations providing an efficient and quick way of calculation. This scope of this chapter was to see if the multidimensional matrix approach could also be applied to capacity planning.

The good results obtained with direct labour capacity calculation permits us to extend this methodology to efficiently calculate also the indirect labour needed, machine saturation calculations etc. We will not go into details since this could essentially be a repetition of what we already saw in detail.

Since the capacity calculation should follow the MPS and MRP, all known ERP systems use a batch mode to calculate the capacity since the input data in turn is updated in batch mode. Combining together all the three approaches we discussed earlier, we can quickly calculate the requirements using matrices. Moreover the matrix structure permits local updates of the DL CRP matrix instead of running the entire calculation from start to end. This can further increase the updating speed. Just like the earlier discussions, we can also have a history of past matrices and a simulation for future as matrices, allowing us to create a sort of time machine. We can also implement algorithms for scenario simulation and analysis. The numeric matrices also allows us to use matrix manipulation formulas and algorithms and a quantitative analysis.

The Direct Labour Capacity Matrix represents how the direct labour requirements will evolve over time, we can approach this system also from a model predictive control approach, taking the periods most valid from the company set time horizon and use it to model the appropriate control signals which in our case are the number of persons present in all the production units.

Chapter 7.

MPC in Financial Planning

The approach of modelling and the techniques are useful also in other sectors of any industry and not just to supply chain. Often the signal theory and techniques developed are confined to pure engineering applications or signal analysis applications. The natural question that we raise is why cannot we use already consolidated theories and techniques used in control theory to other non-control theory areas.

One of the first application could be in finance and accounting. We will focus on the possibility of applying system modelling techniques to model the financial statements. For any business there are three fundamental financial statements: the income statement, balance sheet, and statement of cash flows.

The income statement illustrates the profitability of the company. It begins with the revenue line and after subtracting various expenses arrives at net income [40]. The income statement covers a specified period like quarter or year.

Unlike the income statement, the balance sheet does not account for the entire period and rather is a snapshot of the company at a specific point in time such as the end of the quarter or year [74]. The balance sheet shows the company's resources (assets) and funding for those resources (liabilities and stockholder's equity). Assets must always equal the sum of liabilities and equity.

Lastly, the statement of cash flows is a magnification of the cash account on the balance sheet and accounts for the entire period reconciling the beginning of period to end of period cash balance. It typically begins with net income and is then adjusted for various non-cash expenses and non-cash income to arrive at cash from operating. Cash from investing and financing are then added to cash flow from operations to arrive at net change in cash for the year

Among them we will choose the last one, the cash flow, to apply linear modelling techniques to search for an alternate way to forecast the cash flow. We will approach it from two points of view: from a statistical point of view of distribution fitting and then from signal theory point of view. We will then compare the results to the commonly used averaging method to see which model best fits the data. Before we go into detail, let us briefly examine these three fundamental financial statements.

7.1. Income Statement

The Income Statement is one of a company's core financial statements that shows their profit and loss over a period of time. The profit or loss is determined by taking all revenues and subtracting all expenses from both operating and non-operating activities [99]. The statement displays the company's revenue, costs, gross profit, selling and administrative expenses, other expenses and income, taxes paid, and net profit, in a coherent and logical manner.

The statement is divided into time periods that logically follow the company's operations. These periodic statements are aggregated into total values for quarterly and annual results.

This statement requires the least amount of information from the balance sheet and cash flow statement. Thus, in terms of information, the income statement is a predecessor to the other two core statements. A brief glossary can be found in appendix A.9.

7.2. Balance sheet

A balance sheet is a financial statement that reports a company's assets, liabilities and shareholders' equity at a specific point in time, and provides a basis for computing rates of return and evaluating its capital structure. A balance sheet provides a snapshot of a business' health at a point in time. It is a summary of what the business owns (assets) and owes (liabilities) as well as the amount invested by shareholders at the date of publication. Balance sheets are usually prepared at the close of an accounting period such as month-end, quarter-end, or year-end[12].

The three major components of the balance-sheet are:

- *Assets*: can be defined as the valuables that the company owns to benefit from or are used to generate income. They are the resources of the company that have future economic value. These are categorized into tangible and intangible assets. The tangible assets can be further classified into current, long term and other assets. The non-tangible assets are trademark, copyrights, goodwill etc. Current assets are any asset which can reasonably be expected to be sold, consumed, or exhausted through the normal operations of a business within the current fiscal year or operating cycle (whichever period is longer). Typical current assets include cash, cash equivalents, short-term investments (marketable securities), accounts receivable, stock inventory, supplies, and the portion of prepaid liabilities (sometimes referred to as prepaid expenses) which will be paid within a year [42] [16]. Long term assets or fixed assets refers to assets and property that cannot easily be converted into cash.

- *Liabilities*: refers to debts owed by the business. It could be claims of the creditors against the assets or obligations arising out of past or current transactions. Liabilities are classified into current and long term liabilities. Current liabilities are due within one year portion of long term debt and any other obligations due within a year. Long term liabilities on the other hand are debts that must be repaid in more than one year from the date of the balance sheet.

- *Net worth* (Owner's Equity): is equal to the reported asset minus the reported liability.

Balance sheet is based on the formula:
Assets = liabilities + Net worth

7.3. Statement of Cash Flows

Statement of Cash Flows reports the cash generated and spent during a specific period of time acting as a bridge between the income statement and balance sheet by showing how money moved in and out of the business. They are of fundamental importance in budgeting process[47].

Cash flow in a more direct manner could be considered as flow of cash that is moving in and out of any business. This is a bi-directional flow both in as cash or account payable and out as accounts receivables. If more cash flows in than out then we speak about positive cash flow. The cash need not always be liquidity. A negative cash flow is one the main reasons why business fail.

Cash flow is one of the three main financial statements along with the balance sheet and profit and loss statement. Though the last one is considered as the most important, the cash flow statement is a very critical and integral part of the complete financial statement, often overlooked by non-expert eyes. Together with the other two statements it gives a clear picture of the real financial health of the business. The main thing that we can read from this is the cash that the business really generates.

We can enumerate the importance of cash flow forecasting as follows [79]:

1. Identify potential shortfalls in cash balances in advance, an early warning system of debts;

2. Make sure that the business can afford to pay suppliers and employees;

3. Spot problems with customer payments;

4. An important discipline of financial planning;

5. External stakeholders such as banks may require a regular forecast.

The cash flow statement of a given period states how much cash the business started the period with, how the business spent its cash, how the business received its cash, and the ending cash balance available for future investment or distribution to share holders. The cash flow monitoring better equips management to make informed decisions about regular business operations, the need for further investment in the business, and capital from equity or debt partners. It is vital to make intelligent decisions on financing activities, operating activities and investing activities.

The calculation of the actual amount flowing is a simple plus-minus calculation of the payable and receivable. The variable in this picture is how accurate the actual payment date coincides with the hypothesised payment date. In this chapter we are trying to model the delay variable.

7.4. Modelling the Cashflow Forecasting

In almost all practical business situations, the most used method is the simple average of the past differences between agreed payment date and actual payment date. This value is then added to the ERP (Enterprise Resource Planning) which then simulates the cash flow considering this deterministic value.

One of the main defects of this method is its lack of statistic robustness and extremely high susceptibility to influence by outliers, namely the much larger/much smaller values. In case the individual difference follows a skewed distribution such as a log-normal or Weibull distribution, this simple arithmetic mean has nothing to do with the actual median value which could be a better description of the central tendency. A brief excursus of statistics concepts that we will use could be found in appendix A.1.

The method we propose consists of a deeper data analysis using two different perspectives: Statistical path of probability distribution fitting and from signal processing point of view in modelling the system. We will then combine the two results and then compare the obtained forecast with the average method commonly used with respect to the real data.

The raw data is almost always "dirty", with outliers due to a large variety of reasons which can alter the data like in averaging methods. These can be broadly classified into data errors and process errors. The data errors arise from incorrect data processing like mistyped numbers while the process errors are incorrect data originated correctly from process flaws. One example could be an invoice issued but sent to the customer with a delay, without correcting the due date. There is no clear cut way to define outliers. One of the common methods is to create a Pareto chart for an ABC classification [18] [8], discussed briefly in appendix A.7, and then use only the class A items. Another simpler way is to view the graph to identify the range visually.

We should also take into consideration the usefulness of data. Usually more useful information is provided by the recent data from the previous year or the one prior to it. Older data may be based on an underlying situation which is no longer valid.

7.4.1. Signal Theory Approach

Autoregressive Model (AR)

Autoregressive model is used to represent random time varying processes, which specifies the linear dependence of output variables on the previous values and a non perfectly predictable stochastic term. When we speak of an autoregressive model of order p, we define it as

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \tag{7.1}$$

where c is a constant $\varphi_1, ..., \varphi_p$ are the model parameters and ϵ_t the white noise. The coefficients can be estimated using Least Squares Method [102] or the Method of Moments [25].

AR(0) is the simplest of the AR processes with no dependence between the terms [102]. In AR(1) with positive φ, the previous term along with the noise term contributes to the process output. AR(2) has the two previous terms contributing to the output along with the noise term. If both φ_1 and φ_2 are positive then we have an output that looks like a low pass filter, with a reduction of higher frequency part of the noise. If φ_1 is positive and φ_2 is negative then the output oscillates.

Moving-Average Model (MA)

Moving-Average model is used to represent univariate time series, which specifies the linear dependence of output variables on the previous values and a non perfectly predictable stochastic term. We can thus speak of a linear regression of current value against current and previous white noise error terms.

When we speak of an moving-average model of order q, we define it as

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^{p} \theta_i \varepsilon_{t-i} \tag{7.2}$$

where μ is the mean $\theta_1, ..., \theta_q$ are the model parameters and $\varepsilon_t, ..., \varepsilon_{t-q}$ are the white noise terms.

Unlike the AR model, the white noise error terms are propagated to the future values of the time series directly. In AR model this propagation is indirect. The error terms affect the X values only for the current period and q periods into the future while in case of AR model, this goes infinitely to the future. Moreover since it is not possible to observe the lagged error terms, iterative non-linear fitting procedures are to be employed in place of least squares [6].

Auto Regressive Moving Average Model (ARMA)

ARMA(p,q) refers to the model with p autoregressive terms and q moving-average terms, a combination of AR(p) and MA(q) models[100].

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{p} \theta_i \varepsilon_{t-i} \tag{7.3}$$

The error terms ε_t are generally assumed to be independent identically distributed random variables sampled from a normal distribution ε_t $N(0, \sigma^2)$.

7.5. Cash Flow Forecasting Using Traditional Methods

To illustrate the method we propose, let us analyze a single customer of an automotive component manufacturer, one that has significant number of invoices in the last 5 years. The only criterion for the choice of this particular customer was the high number of invoices and related payments. We consider the data concerning the payments in the years 2012-2017 for identification and the data concerning 2018 to validate the model.

The agreed contractual payment term is 60 days from the date of the invoice. This means that if the goods are shipped today, the invoice is made today and the customer must pay the goods at the latest after 60 days form today.

If they pay a day later, the delay is +1, if they pay 5 days earlier the delay is -5. Since the payment terms can be considered as a fixed shift of the whole scenario by 60 days into the future and since the variable modelled (delay) has nothing to do with this shift, we just consider the delay as if the payments should be done at time $t = 0$. So we will write $x(t)$ instead of $x(60 - t)$ for a payment done with a delay of t days.

Analysing the data we see that the data set is quite widely distributed as shown in figure 7.1.

Let us create the Pareto chart that allows to see the situation more clearly as in figure 7.2.

We can thus narrow down our range to the period between -90 days and +20 days delay, which form the Class A items. Comparing it with the graph of full data set in figure 7.1, we see that there are some values between +20 and +25 that could be significant and a lot of values less than -50 that could be less significant. We can thus empirically choose the differences between the due date and actual payment date in the range from -50 to + 25 for our data analysis, as in figure 7.3.

Let us apply the method of averages, which provides the values shown in the following table.

Year	**Average**	**Std. Dev.**
2012	4.66	8.61
2013	4.82	15.17
2014	5.37	10.38
2015	1.67	13.14
2016	8.04	48.66
2017	**4.09**	18.88
Total	**2.12**	25.88

The main problem with this approach is that it just gives a static value and does not allow any kind of calculation of probabilities with relative confidence level nor does it permit to analyze any if-case scenarios to better plan the cash flow process. We have two values, one double of the other. The only way out in this case is to blindly chose one or other hoping to have made the best choice.

7.6. Cash flow forecasting using distribution fitting method

Let us now take the complete set of data available for the customer and apply the distribution fitting method. We will be using the *Distribution Fitter* App present in the *Statistics and Machine learning Toolbox* of Matlab. The plot of the raw data vs cleaned data is shown in the figure 7.4. A brief excursus of statistics concepts that we will use could be found in appendix A.1.

From the plot we can straight away infer that the distribution could be either Normal or Logistics [44] and we thus plot the raw data, the density and

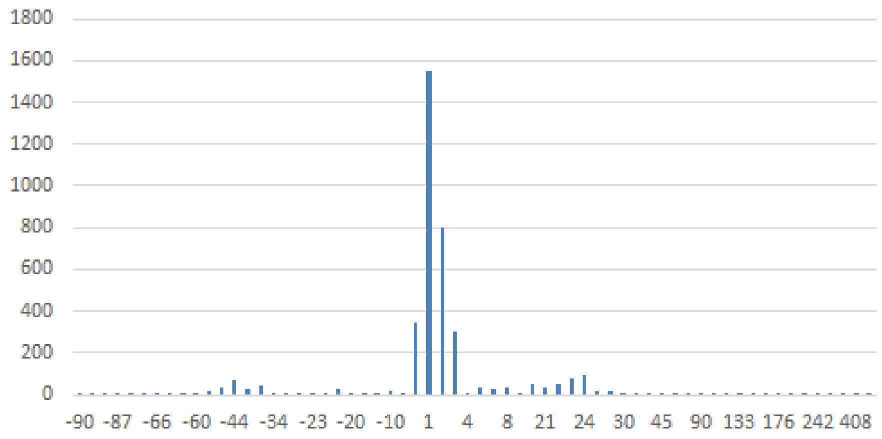

Figure 7.1.: Full data set

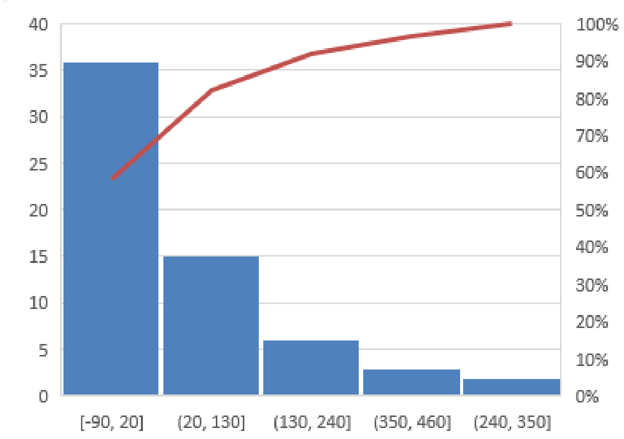

Figure 7.2.: Pareto chart for full data set

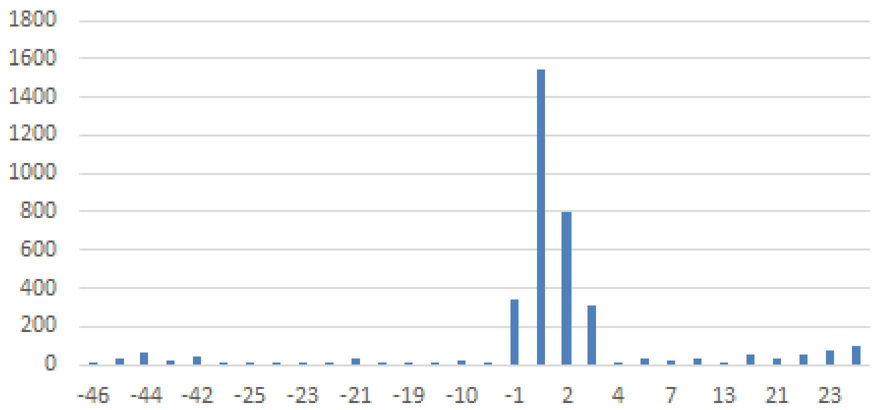

Figure 7.3.: Range of data for analysis

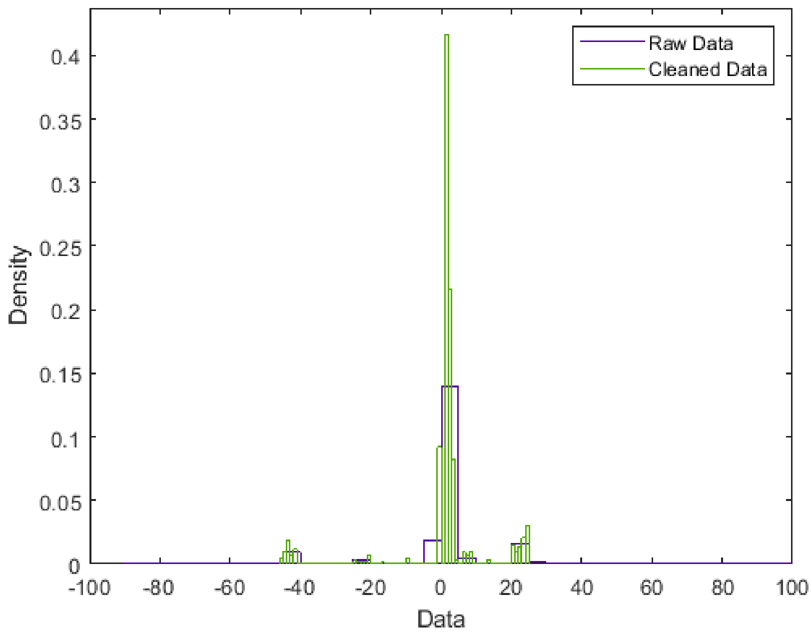

Figure 7.4.: Raw and Cleaned data

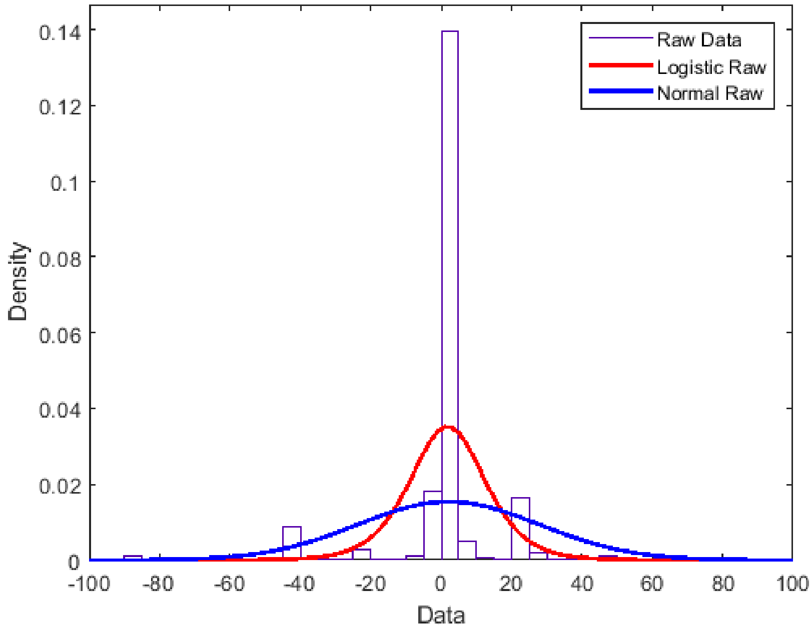

Figure 7.5.: Density of Raw Data vs Logistic and Normal distributions

cumulative distribution function along with those two distributions as shown in figures 7.5 and 7.6.

We can clearly see that the best fitting distribution could be Logistic distribution with mean 1.8465 and variance 165.328. The Normal distribution has the mean 2.12395 and variance 669.861. Thus we can write the probability density function as (7.4) and Cumulative distribution function as (7.5)

$$f(x;\mu,s) = \frac{e^{-\frac{x-\mu}{s}}}{s(1+e^{-\frac{x-\mu}{s}})^2} \tag{7.4}$$

$$f(x;\mu,s) = \frac{1}{1+e^{-\frac{x-\mu}{s}}} \tag{7.5}$$

where μ is the mean, s the scale parameter proportional to the standard deviation, with $x \in (-\infty, \infty)$. The comparison of results with the cleaned data in the delay range from -50 to 25 is shown in figure 7.7 where we can see that the cleaned set of data permits a fit that better follows the data.

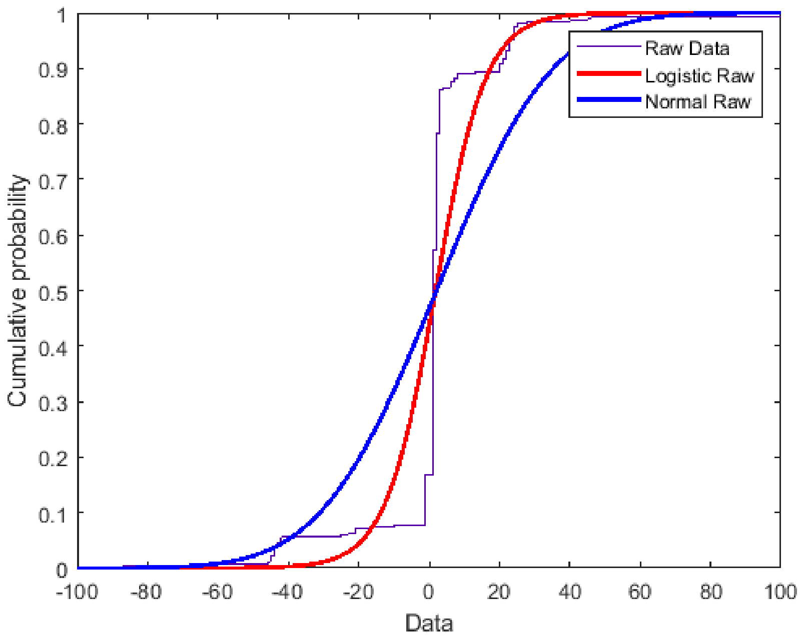

Figure 7.6.: Cumulative Distribution Function of Raw Data vs Logistic and Normal distributions

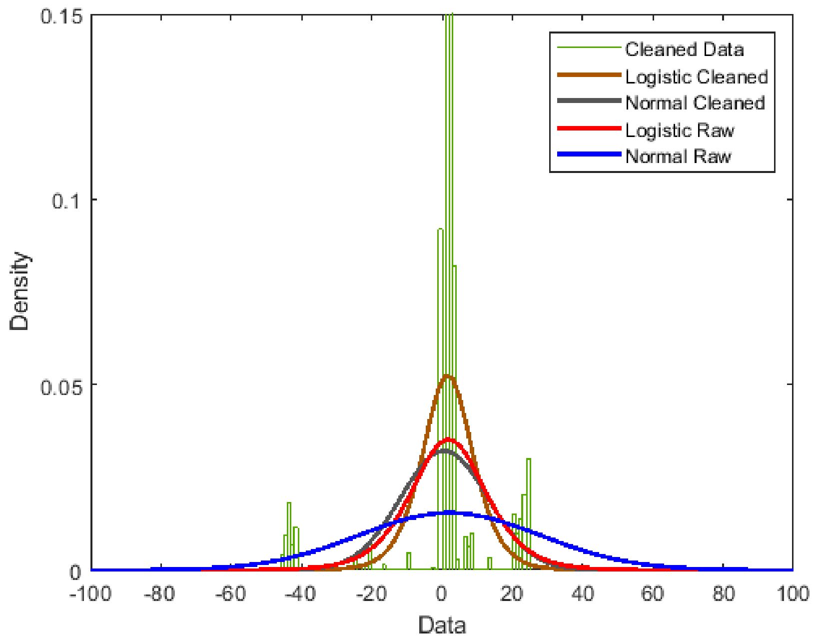

Figure 7.7.: Density plot of cleaned data vs Logistic and Normal distributions

7.7. System Identification

Instead of using the dirty data, which significantly reduces the fit of estimation, we can directly use the most significant segment of delays. For a rapid system identification, we will be using the *System Identification Toolbox* of Matlab. The estimation using the ARMA model did not give any significant results. Using the polynomial model estimation, ARMA(p,q) with p and q ranging from 1 to 4. We can see from the fig 7.4 that there is one main interval [0,4] and two smaller intervals [-45,-35] and [20,30]. To have a better system identification we will treat them separately as three systems. For [0,4] we get the best fit with ARMA(3,3), for [20,30] we get the best fit with ARMA(4,4) and for [-45,-35] we get the best fit with ARMA(1,1). The results are shown as follows:

```
Data [20,30]
Discrete-time ARMA(3,3) model:
Fit to estimation data:77.49%
FPE:0.0259, MSE:0.02579
A(z)y(t) = C(z)e(t)
A(z) = 1 - 0.7206  z^-1 - 0.3703 z^-2
```

```
 + 0.0962  z^-3
C(z) = 1 + 0.2777 z^-1  - 0.0932  z^-2
  + 0.0014 z^-3
Data [20,30]
Discrete-time ARMA(4,4) model:
Fit to estimation data:77.32%
FPE:0.0259, MSE:0.02579
A(z) = 1 - 1.178 z^-1+ 0.8378  z^-2
- 1.285 z^-3 + 0.625 z^-4

 C(z) = 1 - 0.3984  z^-1  + 0.5449  z^-2
   - 0.8715  z^-3 - 0.06693 z^-4

Data [-45,-35]
Discrete-time ARMA(1,1) model:
Fit to estimation data:77.32%
FPE:0.0259, MSE:0.02579

  A(z) = 1 - 0.9982 z^-1
  C(z) = 1 - 0.1036     z^-1
```

Using the model obtained with the real data, the best results were obtained using the ARMA(4,4) model as shown in the figure 7.8. The x-axis shows the time in calendar days and the y-axis shows the delays between the actual payment date and the expected payment date which is just +60 days from the invoice date. We can clearly see that the real values (in violet, with local oscillations) closely follow the predicted values (in black, more linear).

7.8. Comparison with Real Data

We have used two distinct approaches, one statistical and one based on system modelling to predict the payment behaviour of our customer. Let us now compare the result with respect the real data concerning the first semester of 2018.

From the statistical analysis by distribution fitting we arrive at the logistic distribution with a mean of 1.52059 and variance 0.485936. The normal distribution gives a mean of 1.65086 and a variance of 0.765021. The CDF plot is as shown in figure 7.9.

If we need a single value to add to the financial calculations at the company level on the possible delay on the payments from this customer, we could take any of the two values, preferably logistics value of 1.5 days of delay on payments

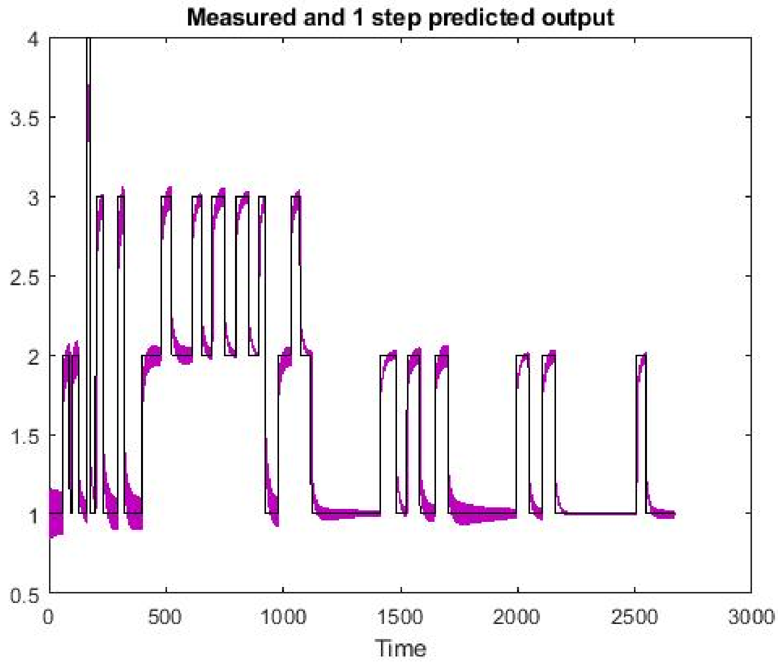

Figure 7.8.: ARMA(4,4)

with 60 calendar days from the date of invoice. We can see that this is quite different from the 4.09 days if we took the average of 2017 or the 2.12 days if we took the 6 year average.

Let us now use the system approach, which in comparison is a continuous approach. There is no single value to be inserted into the system but a set of continuous predicted delay values as shown in figure 7.10, with the `x`-axis showing the calendar days of the time series and `y`-axis the delay between the expected payment date and the real payment date.

We can clearly see that the real values (in violet, with local oscillations) closely follow the predicted values (in black, more linear). with slight delays in some data points, which is acceptable given the nature of data which is directly used from the real world without any kind of cleaning or processing.

This forecasting could be considered a significant improvement over the existing situation. Moreover the possibility of improving the system in an incremental manner with the evolution of new data points makes this approach more interesting.

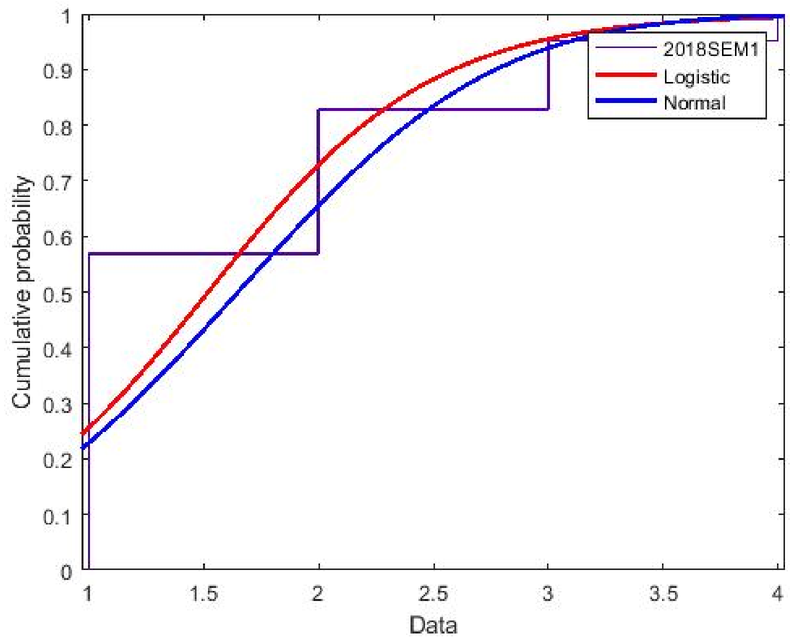

Figure 7.9.: Distribution fitting with new data

7.9. Results

Comparing the average method with the approaches we have presented above, we can see that the system approach and the statistical approach displays a much better performance. The comparison with the data from the first semester of 2018 shows a remarkable coherence with the calculated data as shown in the figures 7.9 and 7.10. The statistical approach could be easier in case a single value is needed, where the continuous projection aspect is lost. Here the system approach come really handy. At any given point we are able to define a linear model from the past data, which predicts the future data with a reasonable and acceptable accuracy. As time flows, we can improve the model with the new information and thus it becomes progressively better at estimating the delay in the actual day of payment, helping a more accurate calculation of the real cash flow. In the case of this customer the daily invoiced amount goes around 10.000 Euro per day, which is an important amount for the company discussed. So having a good way of predicting the modelled data could really mean managing correctly the financial aspects maintaining a good cash flow.

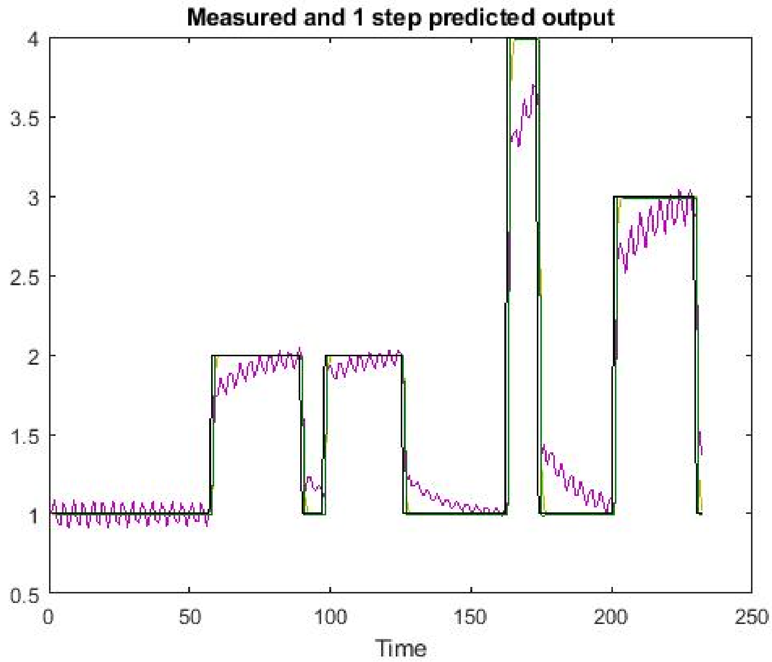

Figure 7.10.: System identification with new data

7.10. Conclusion

This is a first attempt to apply these methods to financial problems. We have used toolboxes available in Matlab without much fine tuning, which could further increase the accuracy of the modelling. Using traditional methods we would have simply used the last year average of 4.09 or the overall average 2.12 which clearly is far from reality.

The actual calculations should also consider the amount of payment related to a single invoice. When the amount is higher as in the case of this customer a two day difference in calculation can make a real difference.

The aim of this study was to check the feasibility of applying the statistical and system modelling approach new to this area and the first results are quite satisfactory. The study will be expanded to more customers with increasing complexity.

This main ideas of this chapter was also briefly presented in an article in International Conference on Control, Automation and Diagnosis (ICCAD) at Grenoble, France [68].

The next step is be to check how the individual analysis of the customers fit the global scenario and how the different systems interact. May be we cannot find a single model always valid for all customers but we could create incremental systems that evaluate the scenario of any customer in a fixed time frame, like 6 months and give an incrementally valid system. Applying the methodology described in this chapter only for the more important customers, we can also reduce the computational load in real life business.

Chapter 8.

Conclusion

The scope of this study was to check the feasibility of applying modelling techniques to supply chain especially to the planning phase, both in the preparation of the Master Production Schedule as well as to the Material Requirements Planning. In this process we saw that an efficient data storing structure and a quick algorithm to easily manipulate that data structure can bring some serious advantages, quickening the process and making a real time elaboration possible.

To correctly place the planning perimeters, we discussed in chapter 2 the main concepts used in demand planning and forecasting. As we saw in the later chapters this is one of the main inputs on which the later systems works on. Following a brief introduction of time series and demand forecasting we made a brief excursus of qualitative demand forecasting techniques, especially on Delphi approach, then we pass on to Quantitative demand forecasting techniques. These techniques are based on past data to forecast the future of the time series with data points indexed in time. We discussed the progression from naive method through averaging methods to end up in different flavours of exponential smoothing. The single exponential smoothing is used to forecast into the future considering just the level. The double exponential smoothing on the other hand considers level and trend and the triple exponential smoothing adds seasonality helping us to predict more data points into the future with better accuracy. We can thus consider the demand as a dynamic time-variant system.

In chapter 3 we discussed in detail the Master Production Schedule (MPS) where we bring down the demand planning from the aggregate product family level to the actual demand for the specific end products. We connect this demand to produce a practical and feasible production plan called as Master Production Schedule. We immediately saw how this can be modelled as system with demand forecast, customer orders (allocated, reserved or unplanned), already planned orders, safety stock and minimum lot sizes as input. The result is a feasible set of planned orders along with a detailed description of net demand, projected available balance and available to promise. The system is derived from classic MPS approaches described in current literature, which we discussed in detail in the first part of chapter 3. After the detailed discussion on how MPS calculation is done traditionally, we saw that this method could further be made quick and efficient by introducing matrices instead of database tables and queries. To represent this statement of volume and timing of end products to be made, we first used a set of individual matrices. We later combined these matrices into a single matrix by adding a third dimension to the already existing axes of the set of items and time horizon to represent the system modelled as in Figure 3.7. We saw a visual representation of this matrix in Figure 3.9. Following MPC methodology only the first time buckets are taken into serious consideration. In any case the number of time buckets to be considered is decided following the company policy.

The next step in production planning is the Material Requirements Planning (MRP) (chapter 4). It calculates the requirement for all the components and raw materials required to produce the finished products as demanded by the Master Production Schedule. From the Master Production Schedule we calculate the net requirement of the components by distributing the demand down through the Bill of Materials by first calculating the gross requirements and then netting stock, planned replenishment and lot sizes to then time-phase this net requirement. In traditional MRP calculations this is done by sequentially applying these operations to all items step by step to all Bill Of Materials levels, until the lowest level is reached. In section 4.3.1 we first saw in detail how this is done with a practical example, modelling it as a system as in Figure 4.1, with the system equations as in Figure 4.2. We made use of MRP table to represent this as described in section 4.2.5.

From a table approach, just as we did with MPS, we verified the feasibility of using matrices to represent the individual system variables and saw that it is possible to join them in a single three dimensional matrix to effectively represent the process. Inspired from the Open Leontief model where the input-output matrix could be used to calculate the gross requirement, we developed an improved set of calculations to use multidimensional matrices as discussed in detail in section 4.4.2.

Following this brief discussion we drew out a model of the MPS from the traditional iterative approach defining the inputs and outputs of the system. From the model we analyzed the possibility of representing the inputs and outputs as matrices along with the system variables. This naturally led us to gather them together into a single matrix of three dimensions. This indicated us a way to get out of the three shortcomings of input-output model, namely include the dynamic form of BOM, considering inventories and lead times in our calculations. As we saw in section 4.4.2, we used sparse and full matrices to shift the single columns of our matrix to include lead times along with inverse discreet Fourier transforms. The five main steps in this calculation are:

1. Calculation of Request Matrix from demand and BOM.
2. Calculating the Gross Requirements, extending the input-output model by considering all dependant demands, minimum lot sizes inventory on hand and scheduled receipts by an up-down strategy of calculation between gross and net demands, subtracting the stock from demand and multiplying it with BOM and Leontief inverse of the BOM as shown in equation 4.10.
3. Calculating the net requirements, planned order receipts and inventory on hand is done from gross requirement calculated at the previous step exactly like the classic MRP approach.
4. Calculating the planned order releases that are to be sent to production facilities or to suppliers.
5. Calculating the work in process that is already in the pipeline.

We saw that this approach drastically reduced the MRP calculation time by the use of a simple and straightforward data structure and a computationally efficient algorithm which uses mathematical frameworks already available. We also have an additional possibility of creating a time machine to do scenario analysis and do local and partial updates instead of calculating everything from the beginning. We can also quickly couple different matrices from different companies to quickly create an efficient EDI system opening also to machine learning algorithms.

In chapter 5 we extended this approach to inventory management by modelling inventory using matrices. Following the process model shown in Figure 5.1 we take the open order list, current stock and existing production/purchasing orders as input to calculate the quantity on hand per item along the time line. Multiplied with average price of items on hand we arrive quickly at the stock value and in case of stock out also the stock out value.

This approach gives a quick and efficient way to detect and identify high inventories quickly pin pointing them to the root cause. We can quickly identify the items that are creating high inventories, the relative cost and under what circumstances this is happening, along with the status before, during and after the event of a high inventory or stock out. We can also estimate the optimal inventory value in an efficient manner, the current evolution and possible future evolution by simulating changes in the input matrices.

In chapter 6 we further extend this method to the operations planning, specifically to Capacity Requirements Planning. From the customer demand exploded down to production demands we calculate the resources the company ought to provide in order to meet the demand. In this chapter we focused specifically on direct labour capacity calculations. As in earlier discussions we approached this from a system point of view as shown in Figure 6.1. From the available direct labour capacity, demand and line efficiency parameters we calculated the direct labour hours required once again using multidimensional matrices.

Rather than an exercise of extending the concepts we developed in previous chapters, the goal of this discussion was to complete the matrix model so that the demand, MPS and MRP calculations are represented in a cohesive and interconnected system. In this way we can do quick, efficient and locally delimited calculations that can then be propagated along these interconnected matrices, using mutually compatible algorithms to end up in direct labour requirements calculations beginning from customer demands.

In chapter 7 we explored the use of modelling techniques discussed earlier to other areas, in this specific case of financial planning which is closely related to the topics discussed earlier. We narrowed down our focus on cash flow forecasting where, after a brief overview of what it is and how it is important, we did a statistical and dynamic system approach side by side to model a possible way of forecasting the cash flow. The traditional methods are vague with a low reliability which does not give any significant information useful to forecast cash flow. We first developed a distribution fitting method using normal and logistic distributions to model a possible strategy.

With system identification method we used the ARMA model and for the sample data we found a best fit of ARMA(4,4). Comparing with subsequent real world data we see that our system gives a fairly reliable model whose output is really close to the observed data as shown in Figure 7.10.

In this thesis, moving along the framework of SIOP and model predictive control strategy we elaborated a new set of tools and data structure. There is still long way to go especially on a stand-alone practical, easily usable, implementation without forgetting to add additional interfaces to further development using AI and machine learning algorithms. The capacity planning side, where we dealt just with direct labor capacity, could also be further extended to indirect labor and machine utilization calculations, which will be continued in further studies.

Appendix A.

Concepts and Definitions

A.1. Random variables and Probability distributions

A random variable is formed by assigning a numerical value to each outcome in the sample space of a particular experiment. The state space of the random variable consists of these numerical values. Technically, a random variable can be thought of as being generated from a function that maps each outcome in a particular sample space onto the real number line $\mathbb{R}$ [61].

Probability distribution is a mathematical function that provides the probabilities of occurrence of the possible outcomes of an experiment. It is a description of a random phenomenon in terms of the probabilities of events. It's a general term to indicate the way the total probability of 1 is distributed over all various possible outcomes (i.e. over entire population).

Probability distributions [75] are of two types:

- *Discrete probability distributions*: the set of possible outcomes is discrete, such as a coin toss or a roll of dice, can be encoded by a discrete list of the probabilities of the outcomes, known as a probability mass function. The distribution of a random variable X is discrete and X is called a discrete random variable, if for all u that runs through the set of all possible values of X, $\sum_u P(X = u) = 1$.

- *Continuous probability distributions*: the set of possible outcomes can take on values in a continuous range (e.g. real numbers), such as the temperature on a given day) is typically described by probability density functions (with the probability of any individual outcome actually being 0). They have a cumulative distribution function that is continuous. The normal distribution is a commonly encountered continuous probability distribution. In this thesis we will focus on 3 continuous probability distributions. The Normal distribution used in relation to real-valued quantities that grow linearly (e.g. errors, offsets), Lognormal distribution used in relation to positive real-valued quantities that grow exponentially (e.g. prices, incomes, populations) and logistic distribution used for modeling growth, and also for logistic regression.

A.2. Mean, Variance and Standard Deviation

The mean is the central tendency of a distribution, normally denoted by μ. Expected value or mean is the weighted average of the possible values, using their probabilities as their weights. In the case of a normal distribution, it defines where the peak is. Most values cluster around the mean.

The Variance σ^2 is the second moment of the PMF (Probability Mass Function) or PDF (Probability Density Function) about the mean; an important measure of the dispersion of the distribution and is the square root of the variance. It is the expectation of the squared deviation of a random variable from its mean. It can be seen as a measure of how far a set of random numbers are spread out from the average of their values.

The standard deviation,normally denoted as σ, measures the variability of the observations. In the case of a normal distribution, it defines the width of the bell curve. The standard deviation determines how far away from the mean the values tend to fall thus representing the typical distance between the observations and the average.

$$Var(X) = E\left[(X - \mu)^2\right] \tag{A.1}$$

The variance can also be thought of as the covariance of a random variable with itself [95]:

$$Var(X) = Cov(X, X) \tag{A.2}$$

The mean and standard deviation are parameter values that apply to the whole populations. Since it's generally impossible to measure them for the whole population, we can use random samples to calculate estimates of these parameters. These are denoted using $\bar{x}$ for the sample mean and s for the sample standard deviation.

A.3. Normal Distribution

One of the mostly used statistical distributions in supply chain is the normal distribution. It has the following characteristics:

- It is a symmetric distribution with no skewness of any kind;
- Has the same mean, median and mode;

- Half of the population is less than the mean;
- Most of the observations cluster around the central peak and the probabilities for values further away from the mean taper off equally in both direction;
- Extreme values in both tails of the distribution are similarly unlikely.

For a normal distribution, the standard deviation is important as it gives the proportion of the values that fall within a specified number of standard deviations from the mean. For example, in a normal distribution, 68% of the observations fall within +/- 1 standard deviation from the mean.

The Normal distribution is shown in figure A.1. This distribution is useful because of the central limit theorem. This states that if a set of independent random variables is obtained and each has the same distribution with mean μ and variance σ^2, then their average always has mean μ and variance $\frac{\sigma^2}{n}$, and their average is normally distributed if the individual random variables are normally distributed.

The central limit theorem provides an important extension to these results by stating that regardless of the actual distribution of the individual random variables X_i , the distribution of their average $\bar{X}$ is closely approximated by a $N(\mu, \sigma^2/n)$ distribution. In other words, the average of a set of independent identically distributed random variables is always approximately normally distributed. The accuracy of the approximation improves as n increases and the average is taken over more random variables. As the sample size increases, the sampling distribution of the mean follows a normal distribution even when the underlying distribution of the original variable is non-normal [33].

Let $X_1, ..., X_n$ be a sequence of independent identically distributed random variables. Suppose that these random variables have an expectation and variance $E(X_i) = \mu$ and $Var(X_i) = \sigma^2$ If

$$\bar{X} = \frac{X_1, ..., X_n}{n} \tag{A.3}$$

then

$$E(\bar{X}) = \mu \tag{A.4}$$

and

$$VAR(\bar{X}) = \frac{\sigma^2}{n} \tag{A.5}$$

and if $X_i \sim N(\mu, \sigma^2)$, then

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n}) \tag{A.6}$$

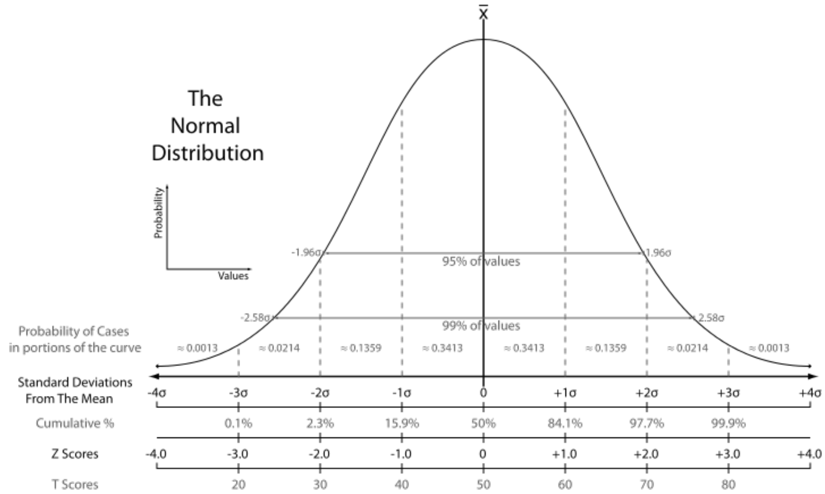

Figure A.1.: Normal distribution

The figures A.2 and A.3 gives an idea of how the curve varies on varying mean and standard deviation.

The probability density of the normal distribution is given by:

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{A.7}$$

where μ is the mean or expectation of the distribution σ the standard deviation, and σ^2 the variance.

The Cumulative distribution function of the normal distribution is given by

$$\frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \tag{A.8}$$

if

$$X \sim N(\mu, \frac{\sigma^2}{n}) \tag{A.9}$$

then the transformed random variable Z such that

$$Z = \frac{X - \mu}{\sigma} \tag{A.10}$$

has a standard normal distribution. This means that any normal distribution

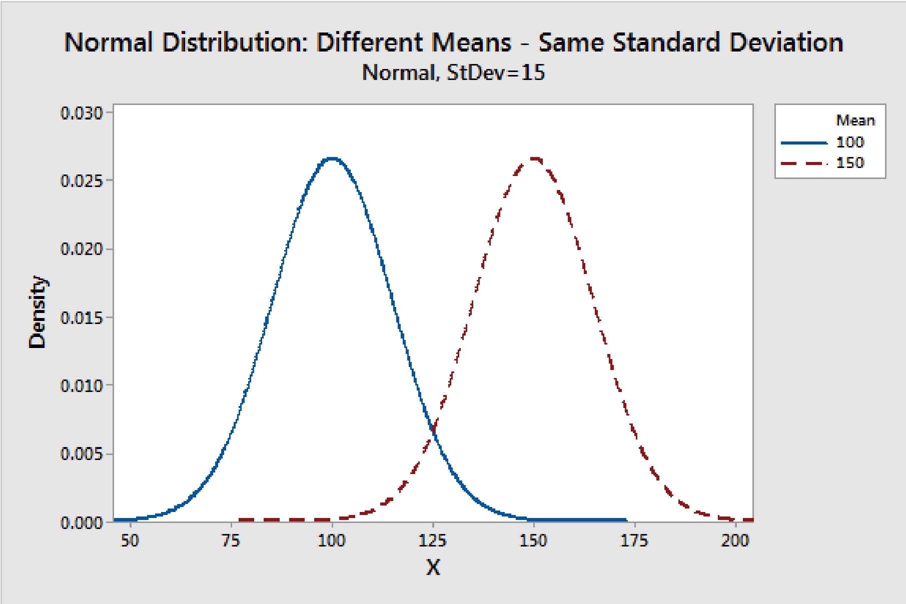

Figure A.2.: Normal distribution with different means

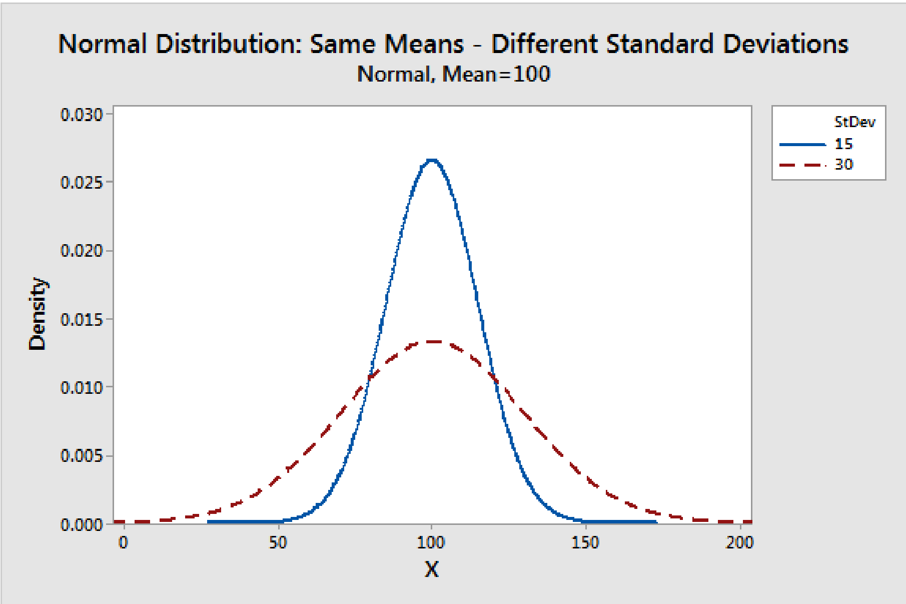

Figure A.3.: Normal distribution with different standard deviations can be related to the standard normal distribution by appropriate scaling and location changes. The random variable Z is known as the *standardized* version of the random variable X. This result implies that the probability values of a general normal distribution can be related to the cumulative distribution function of the standard normal distribution $\Phi(x)$ through the relationship

$$P(a \leq X \leq b) = \Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma}) \tag{A.11}$$

Standard scores are helpful in understanding where a specific observation falls relative to the entire distribution. They also allows us take observations drawn from normally distributed populations that have different means and standard deviations and place them on a standard scale. This standard scale enables us to compare observations that would otherwise be difficult and calculate probabilities across different populations.

The proportion of the area that falls under the curve between two points on a probability distribution plot indicates the probability that a value will fall within that interval. We can transform the values from any normal distribution into Z-scores, and then use a table of standard scores to calculate probabilities.

One of the most important application of normal distribution in supply chain is to calculate the safety stock as detailed in the next section.

Safety Stock and Reorder Point

Assuming that demand during successive unit time periods are independent and identically distributed random variables drawn from a normal distribution, the safety stock can be calculated as

$$SS = z_\alpha \times \sqrt{E(L)\sigma_D^2 + (E(D))^2\sigma_L^2} \tag{A.12}$$

assuming demand during successive unit time periods are independent and identically distributed random variables drawn from a normal distribution. [38]

- α is the service level, z_α is the inverse distribution function of a standard normal distribution with cumulative probability α. Eg. z_α=1.65 for 95% service level.
- $E(L)$ and σ_L are the mean and standard deviation of lead time.
- $E(D)$ and σ_D are the mean and standard deviation of demand.

and reorder point is calculated as

$$ROP = E(L) \cdot E(D) + SS \tag{A.13}$$

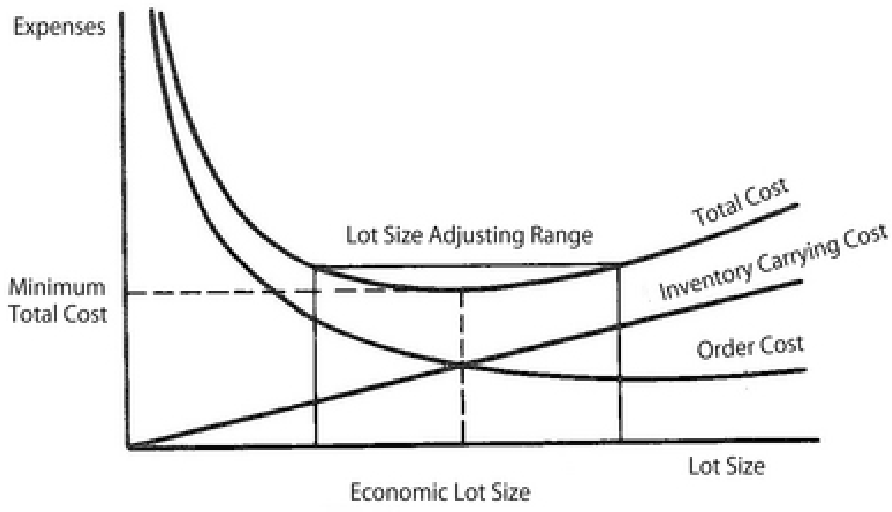

Figure A.4.: Lot Sizing

Minimum, Maximum, Economic Lot sizing

The optimal order quantity is determined by the trade off between the order cost for the items needed and the inventory carrying costs as in figure A.4:

This order quantity is generally called lot or lot size[73]. Economic Lot Size on the other hand refers to the best lot quantity to make the total cost minimum by considering the balance between ordering cost and inventory carrying cost. Lot Size is also called Economic Order Quantity.

A.4. Lognormal Distribution

There are many derived distributions from normal distributions and lognormal is one among them. The lognormal distribution has a positive state space and can be used to model response times and failure times as well as many other phenomena [61].

A random variable X has a lognormal distribution with parameters μ and σ^2 if the transformed random variable $Y = ln(X)$ has a normal distribution with mean μ and σ^2

$$Y = ln(X) \sim N(\mu, \sigma^2) \tag{A.14}$$

The probability density function of X is as follows, graphically as in figure

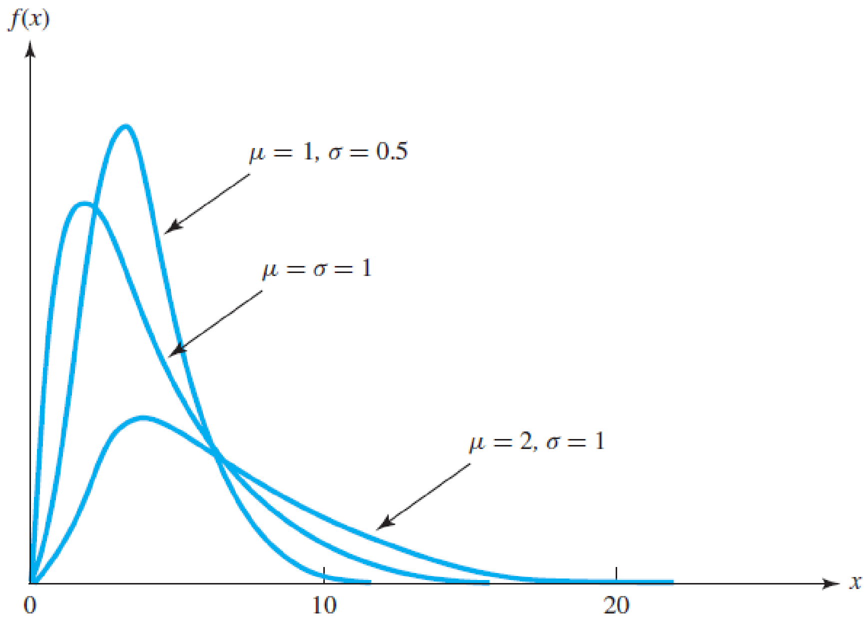

Figure A.5.: PDF of Lognormal Distribution

A.5:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}} \tag{A.15}$$

for $x \geq 0$ and $f(x) = 0$ elsewhere, and the cumulative distribution function is

$$F(x) = \phi\left(\frac{ln(x) - \mu}{\sigma}\right) \tag{A.16}$$

A lognormal distribution has expectation and variance

$$E(X) = e^{\mu + \frac{\sigma^2}{2}} \tag{A.17}$$

$$Var(X) = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1) \tag{A.18}$$

A.5. Logistics Distribution

The logistic distribution is used for modeling growth, and also for logistic regression. It is symmetrical and unimodal. It resembles the normal distribution in shape but has heavier tails (higher kurtosis) which often increases the robustness of analyses based on it compared with using the normal distribution

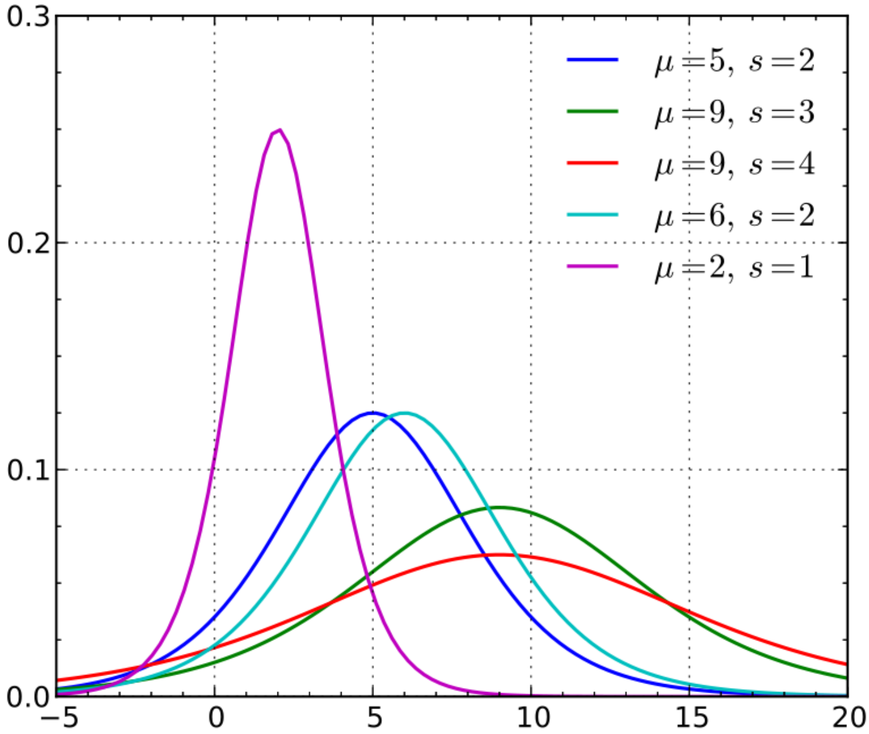

Figure A.6.: PDF of Logistic Distribution

[61]. The logistics distribution has the probability density function (fig. A.6 as:

$$f(x; \mu, s) = \frac{e^{-\frac{x-\mu}{s}}}{s(1 + e^{-\frac{x-\mu}{s}})^2} \tag{A.19}$$

and Cumulative distribution function as:

$$f(x; \mu, s) = \frac{1}{1 + e^{-\frac{x-\mu}{s}}} \tag{A.20}$$

where μ is the mean, s the scale parameter proportional to the standard deviation, with $x \in (-\infty, \infty)$. The logistic distribution is mainly used because the curve has a relatively simple cumulative distribution formula to work with [50].

A.6. Probability Distribution fitting

In probability distribution fitting we analyse the observed data to infer the best possible probability distribution that can be better represent the observed data. The common methods used fall under the two broad categories of parametric methods and regression methods [15]. In practice we a set of observations$x_1, X_2, ..., x_n$ from a sample of unknown population and compare it to check the closeness to a known population with a known probability distribution function.
The four normal steps in fitting distributions are [78]

1. Model/function choice, where the families of distributions are hypothesised;
2. Parameter estimation;
3. Fit quality evaluation;
4. Statistical tests of goodness of fit.

A.7. Pareto Chart

Pareto chart is a type of chart often used in industries which has both line graphs and bar graphs where individual values are represented in descending order by bars while the cumulative total is represented by the line [18] [8]. It is often considered as one among the seven basic tools of quality [90]. The purpose of the Pareto chart is to highlight the most important among a large set of factors, a statistical tool that graphically demonstrates the Pareto principle or the 80–20 rule. The basic principle behind Pareto Principle named after the Italian engineer and economist Vilfredo Pareto, is that almost 80% of the effects come from 20% of the causes.

A.8. KOSU

Kosu, meaning " Manual Time", is a key indicator in production control and is a measure of productivity. It refers to the number of man hours it takes to produce one unit of a product, or to complete a process [45]. The number of workers involved in making a product is multiplied by the number of hours used, which gives the total man hours. This is then divided by the total number of units produced during that time. The lower the kosu the better the productivity.

$$KOSU = \frac{Number\,of\,People \cdot Hours\,Worked}{Number\,of\,parts\,produced} \tag{A.21}$$

For example: if on an 8 hour shift 6 people produced 48 parts then KOSU is:

$$KOSU = \frac{6 \cdot 8}{48} = 1 \tag{A.22}$$

if on an 8 hour shift 6 people produced 60 parts then KOSU improves and isis:

$$KOSU = \frac{6 \cdot 8}{60} = 0.8 \tag{A.23}$$

A.9. Glossary of Financial Terms

Following are brief definition of financial terms used in our discussions [10]:

- Allowance for Bad Debts: Amount of estimated debt to the business that is not expected to be repaid and is subtracted from accounts receivable on the balance sheet. It is also known as an allowance for doubtful accounts.
- Assets: Anything that a business owns that has monetary value.
- Accounts Payable: Debts of the business, often to suppliers, and generally payable within 30 days.
- Accounts Receivable: An amount owed to the business, usually by one of its customers, as result of the extension of credit.
- Accrued Payroll Taxes: Taxes payable for employee services received, but for which payment has not yet been made.
- Balance Sheet: A financial statement showing the assets, liabilities, and net worth of a business as of a specific date.
- Current Assets: Cash and other assets readily converted into cash. Includes accounts receivable, inventory, and prepaid expenses.
- Current Liabilities: The debts of a company which are due and payable within the next 12 months.
- Current Ratio: Current assets divided by current liabilities.
- Debt/Worth Ratio: Total Liabilities divided by Net Worth.
- Depreciation: An accounting convention to take into account the physical deterioration of an asset. It is a systematic method to allocate the historical cost of the asset over its useful life.

- Fixed Assets: Also called long-term assets with a relatively long life that are used in the production of goods and services, rather than being for resale.
- GAAP: Abbreviation of Generally Accepted Accounting Principles. Conventions, rules, and procedures that define accepted accounting practice.
- Inventory: Goods held for sale, raw material and partially finished products which will be sold when they are finished.
- Liabilities: Debts of the business.
- Liquidity: The ability to produce cash from assets in a short period of time.
- Long-Term Liabilities - Debts of a company due after a period of 12 months or longer.
- Net Worth or Owners Equity: The business owner's equity in a company as represented by the difference between assets and liabilities.
- Quick Ratio: Current Assets minus Inventory, divided by Current Liabilities. Also known as the acid test.
- Working Capital: Current Assets minus Current Liabilities.

Appendix B.

MCodes

B.1. MPS Calculation using Multidimensional Matrices

```
%Inputs
TIMEFENCE = 2;
PART_NUMS = 5;
PLANNING_HORIZON =10;
INVENTORY= [55;100;6;20;55];

%Load MPS Matrix
load('MPS.mat');

%Create matrices for ND, PO, PAB and ATP
MPS(:,:,5) = zeros(PART_NUMS, PLANNING_HORIZON+1);
MPS(:,:,7) = zeros(PART_NUMS, PLANNING_HORIZON+1);
MPS(:,:,8) = zeros(PART_NUMS, PLANNING_HORIZON+1);
MPS(:,:,9) = zeros(PART_NUMS, PLANNING_HORIZON+1);

%Calculate Total Orders
MPS(:,:,5) = max(MPS(:,:,1), MPS(:,:,2)
+  MPS(:,:,3) + MPS(:,:,4));
%Clean PO Matrix
MPS(:,1,7) = zeros(PART_NUMS,1);

%Consider Inventories
MPS(:,1,8) = INVENTORY;

for loop = TIMEFENCE:PLANNING_HORIZON+1
%Calculate Net Demand
MPS(:,loop,7) = MPS(:,loop,10)
+ MPS(:,loop,5)
- MPS(:,loop,6)
- MPS(:,loop-1,8);
```

```
%Remove Zeros from PO
TMP = MPS(:,:,7);
TMP(TMP<0)=0;
MPS(:,:,7)=TMP;

%Consider Minimum Lot Size
TMP2 = ceil(bsxfun(@rdivide,MPS(:,:,7),MPS(:,loop,11)));
MPS(:,:,7) = diag(MPS(:,loop,11)) * TMP2;

%Calculate PAB
MPS(:,loop,8) = MPS(:,loop-1,8) + MPS(:,loop,6)
+ MPS(:,loop,7) - MPS(:,loop,5);

%Calculate ATP
if loop <= TIMEFENCE
MPS(:,loop,9) = MPS(:,loop-1,8) + MPS(:,loop,7)
- min( MPS(:,loop,5),
MPS(:,loop,2)
+ MPS(:,loop,3));
else
MPS(:,loop,9) =  MPS(:,loop,7)
- min( MPS(:,loop,5),
MPS(:,loop,2)
+ MPS(:,loop,3));
end
end
TMP = MPS(:,:,9);
TMP(TMP<0)=0;
MPS(:,:,9)=TMP;
```

B.2. MRP Calculation using Multidimensional Matrices

```
clearvars;



% BOM
BOM(:,:,1)=[0,2,1,0,0,0;0,0,0,1,1,0;0,0,0,0,0,2;
            0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];
BOM(:,:,2)=[0,2,1,0,0,0;0,0,0,1,1,0;0,0,0,0,0,2;
            0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];
BOM(:,:,3)=[0,2,1,0,0,0;0,0,0,1,1,0;0,0,0,0,0,2;
```

```
            0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];
BOM(:,:,4)=[0,2,1,0,0,0;0,0,0,1,1,0;0,0,0,0,0,2;
            0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];
BOM(:,:,5)=[0,2,1,0,0,0;0,0,0,1,1,0;0,0,0,0,0,2;
            0,0,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];

%Demand
DEMAND(:,:,1) = [0,0,0,0,0,0];
DEMAND(:,:,2) = [100,0,0,0,0,0];
DEMAND(:,:,3) = [100,0,0,0,0,0];
DEMAND(:,:,4) = [100,0,0,0,0,0];
DEMAND(:,:,5) = [100,0,0,0,0,0];

% Lead Time
LT = [1,1,1,1,1,1];

% Scheduled Receipts
SR(:,:,1) = [0,0,0,0,0,0];
SR(:,:,2) = [0,0,0,0,0,0];
SR(:,:,3) = [0,0,0,0,0,0];
SR(:,:,4) = [0,0,0,0,0,0];
SR(:,:,5) = [0,0,0,0,0,0];

% Minimum Lot Size
MLS(:,:,1) = [1,150,100,1,1,1];
MLS(:,:,2) = [1,150,100,1,1,1];
MLS(:,:,3) = [1,150,100,1,1,1];
MLS(:,:,4) = [1,150,100,1,1,1];
MLS(:,:,5) = [1,150,100,1,1,1];

% Inventory
INV(:,:,1) = [0,500,50,50,0,500];
INV(:,:,2) = [0,0,0,0,0,0];
INV(:,:,3) = [0,0,0,0,0,0];
INV(:,:,4) = [0,0,0,0,0,0];
INV(:,:,5) = [0,0,0,0,0,0];

% Get dimensions of the matrix
[nrows,ncols,ndepth]=size(BOM);

% Initialize matrices
POR=zeros(ndepth,ncols);
POH=zeros(ndepth,ncols);
POX=zeros(ndepth,ncols);
WIP=zeros(ndepth,ncols);
```

```
NR=zeros(ndepth,ncols);
IOH=zeros(ndepth,ncols);

%Start with initial inventories
IOH(1, :) =INV(:,:,1);

%Create Request matrix

%Create Requirement, test upper triangularity
REQUEST=zeros(nrows,ncols,ndepth);
for nloop = 1:ndepth
   RMATRIX = eye(ncols)-BOM(:,:,nloop);
   REQUEST(:,:,nloop) = inv(RMATRIX);
end

% Calculate Gross Requirements
for loop = 2:ndepth
    %Requirement from stock
    PORInventory=zeros(1,ncols);
    PORInventory = INV(:,:,loop-1)*REQUEST(:,:,loop);
    %Requirement from demand
    PORDemand=zeros(1,ncols);
    PORDemand = DEMAND(:,:,loop)*REQUEST(:,:,loop);
    %Requirement from demand - Requirement from stock
    PORDIFF = PORDemand - PORInventory;
    %Remove negatives
    PORDIFF(PORDIFF<0)=0;
    POR(loop,:) = PORDIFF;
    %Save Gross Requirements
    GR(loop, :) = POR(loop, :) * BOM(:,:,loop);
    %Calculate Multiples
    TEMP3 = ceil(bsxfun(@rdivide,POR(loop, :),MLS(:, :, loop)));
    POR(loop, :) =  TEMP3 * diag(MLS(:, :, loop)) ;
    % Updating new stock
    IOH(loop, :) = IOH(loop-1, :) +
                    SR(:, :, loop) +
                   POR(loop, :)
             - GR(loop, :);
    INV(:,2:ncols,loop) = IOH(loop, 2:ncols);

end


%Calculate NR, POR and IOH
IOH(1, :) =INV(:,:,1);
```

```
for loop = 2:ndepth
    %Calculate POH
    POH(loop, :) = max(POH(loop-1, :) + SR(:, :, loop) - GR(loop, :) ,0);
    %Calculate NR
    TEMP = GR(loop, :) - SR(:, :, loop)-POH(loop-1, :);
    TEMP(TEMP<0)=0;
    NR(loop, :) = TEMP;
    %Calculate POR updated
    TEMPPOR= IOH(loop-1, :) + SR(:, :, loop) - GR(loop, :);
    TEMPPOR(TEMPPOR>0)=0;
    TEMPPOR(TEMPPOR<0)=1;
    POR(loop, :) = max(MLS(:, :, loop),
         (NR(loop, :) - IOH(loop-1, :)))*diag(TEMPPOR);
    %Calculate IOH
    IOH(loop, :) = IOH(loop-1, :) +
                    SR(:, :, loop) +
                    POR(loop, :) -
                    GR(loop, :);
end

%Remove negatives from IOH
IOH(IOH<0)=0;

%Recalculate the replenishment matrix POX
TEMP = [zeros(max(LT),size(POR,2)); POR];
[nrows,ncols]=size(TEMP);
TEMP2=full(sparse(mod(-LT,nrows)+1,1:
                         ncols,1,nrows,ncols));
POX=round(ifft(fft(TEMP).*fft(TEMP2),'symmetric'));

% Sum all past demands in first time bucket
TEMP3 = cumsum(POX(1:max(LT)+1,:), 1);
POX(max(LT)+1,:) = TEMP3(max(LT)+1,:);
POX = POX(max(LT)+1:nrows,:);

%Calculate WIP
for loop = 2:ndepth
    WIP(loop, :) = WIP(loop-1, :) +
                     POX(loop, :) -POR(loop, :);
end
WIP(WIP<0)=0;
```

B.3. Capacity Planning Multidimensional Matrices

```
% Production Unit 1
N=[20,22,21,23,7,21,23,21,15,21,20];
F=[1,1,1,1,1,1,1,1,1,1,1];
H=[7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5];
U=[0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01];
V=[0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18];
W=[0,0,0,0,0,0,0,0,0,0,0];
K=[750,750,750,750,750,750,750,750,750,750,750];
D=[7330,7277,7273,7251,7202,7322,7263,7317,7253,6853,6853];
P=[0,0,0,0,0,0,0,0,0,0,0];
S=[0,0,0,0,0,0,0,0,0,0,0];
X=[0,0,0,0,0,0,0,0,0,0,0];
Y=[0,0,0,0,0,0,0,0,0,0,0];
Z=[0,0,0,0,0,0,0,0,0,0,0];
O=[0,0,0,0,0,0,0,0,0,0,0];
L=[0,0,0,0,0,0,0,0,0,0,0];

C1 = cat(1,N,F,H,U,V,W,K,D,P,S,X,Y,Z,O,L);

% Production Unit 2
N=[20,22,21,23,7,21,23,21,15,21,20];
F=[1,1,1,1,1,1,1,1,1,1,1];
H=[7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5];
U=[0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.01];
V=[0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15];
W=[0,0,0,0,0,0,0,0,0,0,0];
K=[250,250,250,250,250,250,250,250,250,250,250];
D=[706,632,627,595,527,696,613,688,598,606,606];
P=[0,0,0,0,0,0,0,0,0,0,0];
S=[0,0,0,0,0,0,0,0,0,0,0];
X=[0,0,0,0,0,0,0,0,0,0,0];
Y=[0,0,0,0,0,0,0,0,0,0,0];
Z=[0,0,0,0,0,0,0,0,0,0,0];
O=[0,0,0,0,0,0,0,0,0,0,0];
L=[0,0,0,0,0,0,0,0,0,0,0];

C2 = cat(1,N,F,H,U,V,W,K,D,P,S,X,Y,Z,O,L);

% Production Unit 3
N=[20,22,21,23,7,21,23,21,15,21,20];
F=[1,1,1,1,1,1,1,1,1,1,1];
H=[7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5];
```

```
U=[0.015,0.015,0.015,0.015,0.015,0.015,0.015,0.015,
                                        0.015,0.015,0.015];
V=[0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25];
W=[0,0,0,0,0,0,0,0,0,0,0];
K=[250,250,250,250,250,250,250,250,250,250,250];
D=[1748,1733,1732,1726,1712,1746,1729,1744,1726,1108,1108];
P=[0,0,0,0,0,0,0,0,0,0,0];
S=[0,0,0,0,0,0,0,0,0,0,0];
X=[0,0,0,0,0,0,0,0,0,0,0];
Y=[0,0,0,0,0,0,0,0,0,0,0];
Z=[0,0,0,0,0,0,0,0,0,0,0];
O=[0,0,0,0,0,0,0,0,0,0,0];
L=[0,0,0,0,0,0,0,0,0,0,0];

C3 = cat(1,N,F,H,U,V,W,K,D,P,S,X,Y,Z,O,L);

% Production Unit 4
N=[20,22,21,23,7,21,23,21,15,21,20];
F=[1,1,1,1,1,1,1,1,1,1,1];
H=[7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5];
U=[0.015,0.015,0.015,0.015,0.015,0.015,0.015,0.015,
                                        0.015,0.015,0.015];
V=[0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25];
W=[0,0,0,0,0,0,0,0,0,0,0];
K=[250,250,250,250,250,250,250,250,250,250,250];
D=[2315,2297,2296,2288,2272,2312,2292,2311,2289,1649,1649];
P=[0,0,0,0,0,0,0,0,0,0,0];
S=[0,0,0,0,0,0,0,0,0,0,0];
X=[0,0,0,0,0,0,0,0,0,0,0];
Y=[0,0,0,0,0,0,0,0,0,0,0];
Z=[0,0,0,0,0,0,0,0,0,0,0];
O=[0,0,0,0,0,0,0,0,0,0,0];
L=[0,0,0,0,0,0,0,0,0,0,0];

C4 = cat(1,N,F,H,U,V,W,K,D,P,S,X,Y,Z,O,L);

% Production Unit 5
N=[20,22,21,23,7,21,23,21,15,21,20];
F=[1,1,1,1,1,1,1,1,1,1,1];
H=[7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5,7.5];
U=[0.02,0.02,0.02,0.02,0.02,0.02,0.02,0.02,0.02,0.02,0.02];
V=[0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18,0.18];
W=[0,0,0,0,0,0,0,0,0,0,0];
K=[800,800,800,800,800,800,800,800,800,800,800];
D=[1704,1687,1686,1679,1664,1702,1683,1700,1680,2009,2009];
```

```
P=[0,0,0,0,0,0,0,0,0,0,0,0];
S=[0,0,0,0,0,0,0,0,0,0,0,0];
X=[0,0,0,0,0,0,0,0,0,0,0,0];
Y=[0,0,0,0,0,0,0,0,0,0,0,0];
Z=[0,0,0,0,0,0,0,0,0,0,0,0];
O=[0,0,0,0,0,0,0,0,0,0,0,0];
L=[0,0,0,0,0,0,0,0,0,0,0,0];

C5 = cat(1,N,F,H,U,V,W,K,D,P,S,X,Y,Z,O,L);

% Capacity Matrix
C = cat(3,C1,C2,C3,C4,C5);

[nrows,ncols,ndepth]=size(C);

% Output
for loop = 1:ndepth
   C(9,:,loop)= C(8,:,loop) ./ C(1,:,loop);                     % Pieces per Day
   C(10,:,loop) = (C(8,:,loop).*C(7,:,loop))./3600;
                 % Total Hours
   C(11,:,loop) = C(10,:,loop).*C(4,:,loop);
                 % Scrap
   C(12,:,loop) = (C(10,:,loop) + C(11,:,loop)).*
                               C(5,:,loop);
                 % Inefficiency
   C(13,:,loop) = (C(6,:,loop)./(1-C(6,:,loop))).*
                           (C(10,:,loop)+C(11,:,loop));
                 % Absenteeism
   C(14,:,loop) = C(10,:,loop)+C(11,:,loop)+C(12,:,loop)+C(13,:,loop);
                % Hours Required
   C(15,:,loop) = (C(14,:,loop)./(C(3,:,loop).*
                            C(2,:,loop)))./C(1,:,loop);
                % DL Required
end

DL = sum(C(nrows,:,:),3);
```

B.4. Inventory Planning using Multidimensional Matrices

```
% Production / Purchase orders
I = [ 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 300 35 0 0 0 0 10 5;
```

```
    0 0 0 20 0 0 0 0 0 0;
    3 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 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 55 0 0 0 0 0;
    0 0 0 0 0 120 0 10 0 56;];
% Open Orders
O = [0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 100 24 100 0 360 780 0 50;
    0 0 0 0 0 0 0 41 0 36;
    0 0 0 0 5 5 100 12 0 0;
    0 0 0 0 0 0 0 0 0 0;
    20 0 0 5 5 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 73 0 0;
    0 240 0 0 0 0 0 0 0 0;];
% Inventory matrix
W = [2 3 0 5 35 40 12 980 22 25];
% Fill the rest of the inventory matrix with zeros
W=[W; zeros(size(I,2)-1,size(I,2))];
% Price Vector
P = [2 3 5 5 3 4 2 9 7 2];
% Transaction matrix
X = I - O;
% instantaneous inventory transactions
W = W + X;
% Inventory progression
W = cumsum(W,1);
% Inventory Value Matrix
V = W*diag(P);
V(V<0)=0;
% Calculate the stock value for single time buckets
VT = cumsum(V,2);
VT=VT(:,size(I,2));
% Retain only negative values in stockout
S = W*diag(P);
S(S>0)=0;
% Calculate stockout for single time buckets
ST = cumsum(S,2);
ST=ST(:,size(I,2));
```

Bibliography

[1] Joseph A. Orlicky. «Net Change Material Requirements Planning.» In: *IBM Systems Journal* 12 (Jan. 1973), pp. 2–29. DOI: 10.1147/sj.121.0002.

[2] Dale E. Seborg et al. *Process Dynamics and Control*. fourth. Wiley, Sept. 2016.

[3] F. Roberts Jacobs et al. *Manufacturing Planning and Control for Supply Chain Management*. McGraw-Hill Irwin, 2011.

[4] Jacques Richalet et al. «Model algorithmic control of industrial processes». In: *IFAC Proceedings Volumes* 10.16 (1977), pp. 103–120.

[5] APICS. *APICS Operations Management Body of Knowledge (OMBOK) Framework*. APICS The Association for Operations Management, 2011.

[6] Sandra Arlinghaus. *Practical Handbook of Curve Fitting*. CRC Press, 1994.

[7] J. Scott Armstrong. *Principles of Forecasting. A Handbook for Researchers and Practitioners*. first. Springer US, 2001.

[8] D. C. Whybark B. E. Flores. «Implementing Multiple Criteria ABC Analysis». In: *Journal of Operations Management* 7 (1987), pp. 79–85.

[9] *Birth Death Process*. Accessed: 2019-12-12. URL: https://en.m.wikipedia.org/wiki/Birth-death_process.

[10] E. Bond. *Business Builder 2: How to Prepare and Analyze a Balance Sheet*. Edward Lowe Foundation. Accessed: 2019-05-15. 2005. URL: https://www.zionsbank.com/pdfs/biz_resources_book-2.pdf.

[11] E. Oran Brigham. *The Fast Fourier Transform: An Introduction to Its Theory and Application*. Prentice Hall, 1973.

[12] Horace R. Brock. *Accounting: Principles and Applications*. Ed. by Glencoe/McGraw-Hill School Pub Co. Sixth. 1990.

[13] Lora Cecere. *A Practitioner's Guide to Demand Planning*. https://www.supplychain247.com/article/a_practitioners_guide_to_demand_planning. Accessed: 2010-04-19. July 2014.

[14] CIPS. *Procurement and Supply Operations*. CIPS in partnership with Profex Publishing, 2012.

[15] Harald Cramer. *Mathematical Methods Of Statistics*. Princeton University Press, 1946.

[16] *Current Asset*. Accessed: 2019-12-12. URL: `https://en.wikipedia.org/wiki/Current_asset`.

[17] *Delphi Method*. Accessed: 2019-12-12. URL: `https://www.investopedia.com/terms/d/delphi-method.asp`.

[18] H. F. Dickie. «ABC Inventory Analysis shoots for dollars not pennies». In: *Factory Management and Maintenance* 109 (1951), pp. 92–94.

[19] M. Bixby Cooper Donald J. Bowersox David J. Closs. *Supply Chain Logistics Management*. third. McGraw-Hill, 2010.

[20] Carlos Bordons Alba Eduardo F. Camacho Carlos Bordons. *Model Predictive Control*. Springer London, 2004.

[21] *Evolution of Transportation*. Accessed: 2019-12-12. URL: `https://www.bartleby.com/essay/Evolution-of-Transportation-FKJY6JEK6ZYA`.

[22] *Exponential Smoothing*. Accessed: 2019-12-12. URL: `https://en.wikipedia.org/wiki/Exponential_smoothing`.

[23] J. Freund and B. Rücker. *Real-Life BPMN: Using BPMN, CMMN and DMN to Analyze, Improve, and Automate Processes in Your Company*. CreateSpace Independent Publishing Platform, 2016.

[24] Dave Garwood. *Bills of Material: Structured for Excellence*. Dogwood Publishing Company, 1995.

[25] Walton C. Gibson. *The Method of Moments in Electromagnetics*. Chapman and Hall/CRC, 2007.

[26] M. Gifkins and D. Hitchcock. *The EDI Handbook: Trading in the 1990s*. Blenheim Online, 1988. ISBN: 9780863531484.

[27] Charles F Golub Gene H.; Van Loan. *Matrix Computations*. Johns Hopkins, 1996.

[28] Robert Grubbstrom et al. «Inter-linking MRP theory and production and inventory control models». In: Feb. 2004.

[29] C. Tirtiroglu E. Gunasekaran A. Patel. «Performance measures and metrics in a supply chain environment». In: *International journal of operations & production Management 21, no. 1/2* (2009), pp. 71–87.

[30] Zipkin Paul H. *Foundations of Inventory Management*. McGraw-Hill/Irwin, 2000.

[31] Emmanuel Hadzipetros. *Architecting EDI with SAP IDocs: The Comprehensive Guide*. 2nd. SAP PRESS, 2013.

[32] David Hawkins and Herbert A. Simon. «Note: Some Conditions of Macroeconomic Stability». In: *Econometrica* 17.3/4 (1949), pp. 245–248. ISSN: 00129682, 14680262. URL: `http://www.jstor.org/stable/1905526`.

[33] Anthony Hayter. *Probability and Statistics for Engineers and Scientists*. 4th ed. Brooks/Cole, 1946.

[34] Wilmjakob Johannes Herlyn. *Zur Problematik der Abbildung variantenreicher Erzeugnisse in der Automobilindustrie*. VDI Verlag,Düsseldorf, 1990.

[35] *History of Supply Chain Management*. Accessed: 2019-12-12. URL: `https://cerasis.com/history-of-supply-chain-management/`.

[36] *History of Supply Chain Management*. Accessed: 2019-12-12. URL: `https://www.linkedin.com/pulse/history-supply-chain-management-jsc-engagement-team`.

[37] Charles C. Holt. *Forecasting Trends and Seasonal by Exponentially Weighted Averages*. Office of Naval Research Memorandum, 1957.

[38] Mark L. Spearman Wallace J. Hopp. *Factory Physics 3rd ed.* Waveland Pr Inc, 2011.

[39] M. Imai. *Gemba Kaizen: A Commonsense, Low-Cost Approach to Management*. Mcgraw-hill, 1997.

[40] *Income Statement*. Accessed: 2019-12-12. URL: `https://quizlet.com/195101959/missed-stuff-flash-cards/`.

[41] «Inventory Management». In: *Lecture notes Inventory Management* (2013).

[42] J.E. Goodman J. Downes. *Dictionary of Finance & Investment Terms*. Ed. by 2003 Barons Financial Guides. 2003.

[43] Christopher Gray John Dougherty. *Sales & Operations Planning - Best Practices: Lessons Learned*. Trafford, 2006.

[44] Jessica Hwang Joseph K. Blitzstein. *Introduction to Probability*. CRC Press, 2015, pp. 206–217.

[45] *KOSU*. Accessed: 2019-12-12. URL: `https://www.graphicproducts.com/articles/kosu/`.

[46] Rakesh Kumar and Samir K Srivastava. «A Framework for Improving Sales & Operations Planning». In: *Metamorphosis* 13.1 (2014), pp. 16–25.

[47] William R. Lalli. *Handbook of Budgeting*. Ed. by John Wiley & Sons. Sixth. 2012.

[48] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1999. ISBN: 9780898719734. URL: `https://books.google.it/books?id=YhhkipYCmPIC`.

[49] W. Leontief. *Input-output economics*. Oxford University Press, 1986. ISBN: 9780195035254.

[50] *Logistic Distribution*. Accessed: 2019-12-12. URL: `https://www.statisticshowto.datasciencecentral.com/logistic-distribution/`.

[51] «LOGISTICS SERVICES Concepts and definitions». In: (2019). URL: `http://www.barceloc.com`.

[52] *Matlab Online Help*. Accessed: 2019-06-04. URL: `https://it.mathworks.com/help/matlab/ref/sparse.html?searchHighlight=sparse&s_tid=doc_srchtitle`.

[53] Ronald E. Miller and Peter D. Blair. *Input-Output Analysis: Foundations and Extensions*. 2nd ed. Cambridge University Press, 2009. DOI: `10.1017/CBO9780511626982`.

[54] *Model predictive control*. Accessed: 2019-12-12. URL: `https://en.wikipedia.org/wiki/Model_predictive_control`.

[55] *Model predictive control*. Accessed: 2019-12-12. URL: `https://db0nus869y26v.cloudfront.net/en/Model_predictive_control`.

[56] *Model predictive control*. Accessed: 2019-12-12. URL: `https://www.plantautomation.com/doc/a-simplified-and-integrated-approach-to-model-0001`.

[57] Y. Monden. *Toyota Production System: An Integrated Approach to Just-in-time*. Engineering & Management Press, 1998.

[58] Steven Nahmias. *Production and Operations Analysis*. McGraw-Hill/Irwin, 2009.

[59] Robert Johnson Nigel Slack Stuart Chambers. *Operations Management*. Prentice Hall, 2007.

[60] Hukukane Nikaido. *Introduction to sets and mappings in modern economics*. 1st ed. North-Holland Pub. Co, 1970.

[61] NIST/SEMATECH. *Engineering Statistics Handbook*. National Institute of Standards and Technology, International SEMATECH, 2012. URL: `http://www.itl.nist.gov/div898/handbook/`.

[62] Olaf Helmer Norman Dalkey. «An Experimental Application of the DELPHI Method to the Use of Experts». In: *Management Science* 9.3 (Apr. 1963), pp. 458–467.

[63] Harry Smith Norman R. Draper. *Applied Regression Analysis*. third. Wiley, Apr. 1998.

[64] Rob O'Byrne. *Is Your Company on top of its Inventory Planning Game?* `https://www.logisticsbureau.com`. [Online; accessed 10-May-2019]. 2019.

[65] T. Ohno. *Toyota Production System: Beyond Large-Scale Production*. Taylor & Francis, 1988.

[66] Jos Peeters. «Early MRP Systems at Royal Philips Electronics in the 1960s and 1970s». In: *IEEE Ann. Hist. Comput.* 31.2 (Apr. 2009), pp. 56–69. ISSN: 1058-6180. DOI: `10.1109/MAHC.2009.23`. URL: `http://dx.doi.org/10.1109/MAHC.2009.23`.

[67] P. Pennesi and gestionale e dell'automazione Università politecnica delle Marche. : Dipartimento di ingegneria informatica. *Adaptive Model Predictive Control in the Inventory Control Problem: Dottorato Di Ricerca in Sistemi Artificiali Intelligenti Nell'ingegneria Dell'informazione E Nell'ingegneria Industriale : Tesi Di Dottorato*. 2006.

[68] A.M. Perdon and Kishore Chalakkal Varghese. «Cashflow forecasting with linear models». In: *International Conference on Control, Automation and Diagnosis (ICCAD), Grenoble, France, pp 138-143* (July 2019).

[69] A.M. Perdon and Kishore Chalakkal Varghese. «Multidimensional Matrix Approach to Material Requirements Planning». In: *International Conference on Control, Automation and Diagnosis (ICCAD), Grenoble, France, pp 293-297* (July 2019).

[70] A.M. Perdon and Kishore Chalakkal Varghese. «Stockout prediction using matrices and linear supply chain model». In: *IEEE-2018 International Conference on Control Automation and Diagnosis, ICCAD'1 March 19-21 2018 Marrakech-Morocco* (Mar. 2018).

[71] Annamaria Perdon. «Adaptive Model Predictive Control in the Inventory Control Problem». In: *Lecture notes on Analisi e Controllo di Sistemi Complessi* (2015).

[72] Petrovic R. Petrovic D. Roy R. «Supply chain modelling using fuzzy sets». In: *International Journal of Production Economics* 59 (1999), pp. 443–453.

[73] David J Piasecki. *Inventory Management Explained: A focus on Forecasting, Lot Sizing, Safety Stock, and Ordering Systems*. OPS Publishing, 2009.

[74] *Please walk me through the three financial statements*. Accessed: 2019-12-12. URL: `https://www.wallstreetprep.com/knowledge/please-walk-me-through-the-three-financial-statements/`.

[75] *Probability Distribution.* Accessed: 2019-12-12. URL: https://en.wikipedia.org/wiki/Probability_distribution.

[76] «Production Operations and Logistics Management». In: *Lecture notes MAN 3505* (2019).

[77] John R. Gilbert, Cleve Moler, and Robert Schreiber. «Sparse matrices in MATLAB: Design and implementation». In: *SIAM Journal on Matrix Analysis and Applications* 13 (May 1997). DOI: 10.1137/0613024.

[78] Vito Ricci. «FITTING DISTRIBUTIONS WITH R». In: (2005). URL: https://cran.r-project.org/doc/contrib/Ricci-distributions-en.pdf.

[79] Jim Riley. «Why is the cash flow forecast so important?» In: (2009). URL: https://www.tutor2u.net/business/blog/why-is-the-cash-flow-forecast-so-important.

[80] Arthur D. Little Robert G. Brown. *Exponential Smoothing for Predicting Demand.* Cambridge, Massachusetts, 1956.

[81] Adam Robinson. *The Evolution and History of Supply Chain Management.* Accessed: 2019-07-16. 2019. URL: https://cerasis.com/history-of-supply-chain-management/.

[82] Rushton A., Oxley J., Croucher P. *The Handbook of Logistics and Distribution Management.* Ed. by Kogan page. Second. 2000.

[83] Al Saif. «Introduction to Model Based Predictive Control». In: *Lecture notes CISE 423-162* (2019).

[84] Radim Špicar. «System Dynamics Archetypes in Capacity Planning». In: *Procedia Engineering* 69 (2014). 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013, pp. 1350–1355.

[85] Rob J. Hyndman Spyros G. Makridakis Steven C. Wheelwright. *Forecasting: Methods and Applications.* third. Wiley, Jan. 1998.

[86] *Stockout Costs and Effects.* Accessed: 2019-12-12. URL: https://www.thebalancesmb.com/stockout-costs-and-effects-2221391.

[87] Roland Stoer Josef; Bulirsch. *Introduction to Numerical Analysis 3rd ed.* Springer-Verlag, 2002.

[88] Heeyyoung Kim Sungil Kim. «A new metric of absolute percentage error for intermittent demand forecasts». In: *International Journal of Forecasting* 32 (2016).

[89] *Supply Chain Management.* Accessed: 2019-12-12. URL: https://www.supplychainmanagement.in/supply-chain-management/index.htm.

[90] Nancy R. Tague. *The Quality Toolbox.* 2nd ed. ASQ Quality Press, 2005.

[91] Frederick Winslow Taylor. *The Principles of Scientific Management.* Harper & Brothers, 1911.

[92] *The Evolution of SCL.* Accessed: 2019-12-12. URL: `https://www.scl.gatech.edu/about/scl/history`.

[93] Lluís Pacheco Valls. «Local Model Predictive Control for Navigation of a Wheeled Mobile Robot Using Monocular Information». PhD thesis. Universitat de Girona, June 2009.

[94] C. Van Loan. *Computational Frameworks for the Fast Fourier Transform.* Society for Industrial and Applied Mathematics, 1992.

[95] *Variance.* Accessed: 2019-12-12. URL: `https://en.wikipedia.org/wiki/Variance`.

[96] Kate Vitasek. *CSCMP Supply Chain Management Definitions and Glossary.* Council of Supply Chain Management Professionals. Accessed: 2019-07-16. 2013. URL: `https://cscmp.org/CSCMP/Academia/SCM_Definitions_and_Glossary_of_Terms/`.

[97] A.C. Ward and Lean Enterprise Institute. *Lean Product and Process Development.* Lean Enterprise Institute, 2007.

[98] Waters D. *Quantitative Methods for Business.* Ed. by Addison Wesley. Second. 1997.

[99] *What is the Income Statement?* Accessed: 2019-12-12. URL: `https://corporatefinanceinstitute.com/resources/knowledge/accounting/income-statement/`.

[100] Peter Whittle. *Prediction and Regulation by Linear Least-Square Methods.* University of Minnesota Press, 1983.

[101] Peter R. Winters. «Forecasting Sales by Exponentially Weighted Moving Averages». In: *Management Science* 6.3 (Apr. 1960).

[102] John Wolberg. *Data Analysis Using the Method of Least Squares.* Springer-Verlag Berlin Heidelberg, 2006.