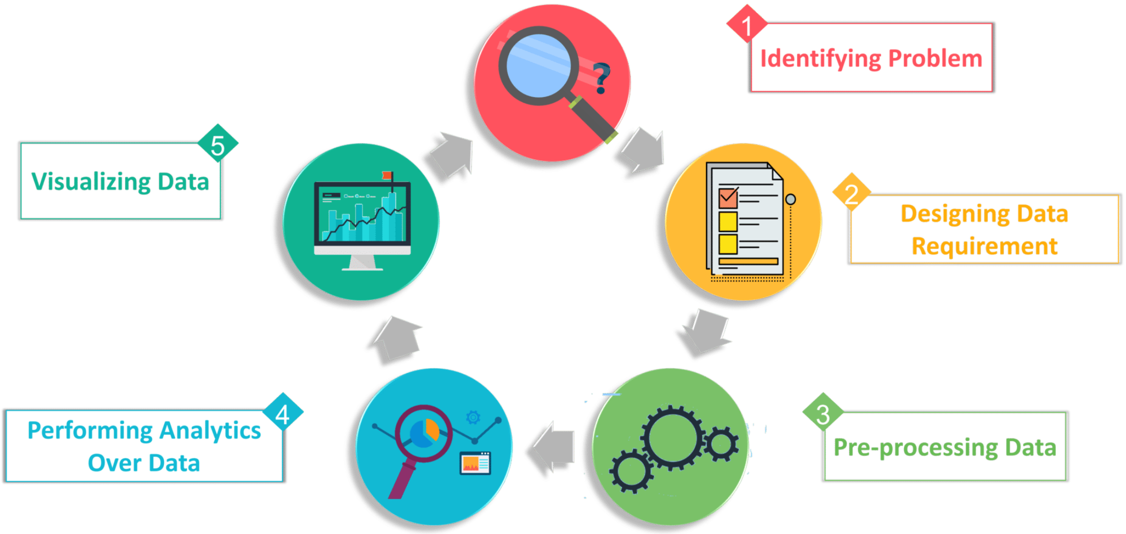
CHALMERS
UNIVERSITY OF TECHNOLOGY



Analytic tool for identifying bottlenecks using Turning Point method

A case study on a Flexible Manufacturing Cell at SKF AB

Master's thesis in Production Engineering

MANOJ SINGH
HASNAIN THATHIA

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Analytic tool for identifying bottlenecks using Turning Point method

MANOJ SINGH
HASNAIN THATHIA

Department of Industrial and Materials Science
Division of Production Systems
Chalmers University of Technology
Gothenburg, Sweden 2019

Analytic tool for identifying bottlenecks using Turning Point method
MANOJ SINGH
HASNAIN THATHIA

Supervisors:
Jan Ek, Department of Process Development, SKF
Mukund Subramaniyan, Department of Industrial and Material Science, Chalmers
Examiner: Anders Skoogh, Department of Industrial and Material Science, Chalmers

Master's Thesis 2019
Department of Industrial and Material Science
Division of Production Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Big data analytics process (Adopted from; https://www.edureka.co/blog/big-data-analytics/)

Gothenburg, Sweden 2019

Analytic tool for identifying bottlenecks using Turning Point method
MANOJ SINGH
HASNAIN THATHIA
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

Manufacturing industry has come a long way ahead since the first industrial revolution. The fourth industrial revolution, Industry 4.0 gives companies an opportunity to make better informed fact-based decisions. Identifying system bottleneck and improving them provides companies the opportunity to utilize their resources more efficiently and be more productive. Productivity and system throughput are two major Key Performance Indicators. It is thus crucial to have an efficient production system that delivers quality products and the desired throughput. In order to achieve this and meet the customer demands in a short lead time, SKF wants to identify the bottleneck machine in a FMC that causes a mismatch between the observed and the desired throughput.

The purpose of the thesis is to improve the productivity/throughput of the production line by facilitating real-time decision making. The aim of the thesis is to demonstrate an approach towards bottleneck detection based on real-time data for a flexible manufacturing system and the Turning point bottleneck detection method was selected. It involves developing a template in Microsoft Excel by identifying important parameters from the data collected to identify the bottleneck machine.

CRISP-DM methodology has been used for data mining purpose. Machine states (working, idle, breakdown, set-up) are a prerequisite for most of the real-time bottleneck detection methods. In this thesis work, an algorithm is developed to calculate machines states from the PLC signals (communication signal between the robot and the machines) which in turn is used to identify bottleneck using Turning point method. The data was cleaned, prepared and analyzed to successfully identify the bottleneck machine. The machine states calculated can be used with different bottleneck detection methods. Analysis is carried out on historical data but the algorithm can be modeled in real-time to identify shifting bottleneck in a FMC.

Keywords: Flexible Manufacturing Cell (FMC), Bottleneck machine, Turning Point Method, Starving time, Blocking time, Machine States, PLC

Acknowledgements

First and foremost, we would like to thank our supervisor Mr. Mukund Subramaniyan for his continuous guidance and feedback throughout the course of this thesis. His knowledge and research work in data-driven bottleneck identification methods was very helpful and it motivated us in exploring new ideas to approach the problem. We would also like to thank our examiner Mr. Anders Skoogh for sharing his expertise on bottleneck detection during the various coursework of our masters program and guidance throughout this thesis work.

We would like to thank our supervisor Jan Ek at SKF AB for his guidance, support and feedback during the thesis. The best part of working with him was the autonomy and flexibility we received in order to proceed with our ideas. A big thank you to Anneli Sundblom for showing faith and trust in us by giving us an opportunity to perform master thesis on a topic based on our interests and future prospects at SKF AB. Lastly, we would like to thank Niklas Borg of ÅF Consult, Mats Dirfors and Robert from SKF AB for their knowledge sharing, guidance and help during the thesis work.

Finally we would like to thank our family and friends for their support throughout the years. Without them, we would never have come so far and achieved a lot.

Manoj Singh, Hasnain Thathia
Gothenburg, August 2019

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

The fourth industrial revolution, also known as Industrie 4.0, is based on the concept of cyber-physical systems and internet technologies applied to industrial production systems (Drath and Horch, 2014; Liao et al., 2017; Monostori, 2014). The future industry will require a strong inter-connectivity between the systems and this can be achieved with the help of the following key elements: (1) Collect the data with the help of smart sensors; (2) Structure and store the data by using an IT system; and (3) Perform analysis with machine learning algorithms to interpret a decision. This collectively is termed as a cyber-physical production system.

Following tremendous advancements in telecommunication and sensor technology, a large amount of data can be gathered, also known as big data. Big data is considered as a driver for innovation and provides companies the opportunities to be more competitive (Tayal et al., 2018; European Commission, 2016). The European Commission (2016) has stated that the amount of data captured is increasing by 40 percent every year and industrial companies are expected to generate cost savings of about 3.6 percent per annum. Massive amounts of data are generated from employees, customers, processes, businesses, products and machines with the help of intelligent sensors (European Commission, 2016; Gölzer and Fritzsche, 2017). Collecting and storing data, however, are insufficient to deliver benefits. It is through the use of smart sensors, powerful analytical tools and complex algorithms such as machine learning, that these data can be analyzed and deliver direct value for performance management and decision-making. For instance, real-time data can be used to analyze suppliers' performance (Tayal et al., 2018), improve the overall equipment efficiency of the machines by using predictive maintenance strategies (Neugebauer et al., 2016).

Real-time data analytics has the potential to improve activities within production by improving cycle-time, reducing set-up time and prioritizing maintenance activities on the bottleneck machines to improve the productivity (Subramaniyan et al., 2016). Thus, the thesis aims at demonstrating an approach towards bottleneck detection for a flexible manufacturing system by performing data analytics in order to increase the overall throughput.

1.2 Problem Statement

SKF has been the world leader in technology especially in areas of bearings, motion technologies, seals and lubrication systems. SKF has been working with Industry 4.0 concepts in their Factory D wherein spherical roller bearings are manufactured. With the aim of creating a digital factory, tremendous amount of data is being generated from their sensors and gathered for the purpose of analysis in order to optimize the production system. The idea is to shift towards making more fact-based decisions.
SKF manufactures several different variants and has a broad product range. Thus, it becomes very important to have an efficient production system that delivers quality products and the desired throughput. In order to achieve this and meet the customer demands in a short lead time, SKF wants to identify the bottleneck machine in a FMC causing a mismatch between the observed and the desired throughput.

1.3 Purpose

The purpose of the thesis is to improve the productivity/ throughput of the production line by facilitating real-time decision making. With the advent of Industry 4.0, manufacturing companies have been capturing tremendous amount of data using sensor technologies. The real-time data gathered from these machines provide us with improvement opportunities such as increase in the overall productivity. This can be achieved by identifying the bottleneck machine/process and making improvements in the bottleneck to increase the throughput and productivity. Thus, SKF would like to improve their throughput in the production system of Factory D by performing valuable analytics on the data gathered from their machines.

1.4 Aim

The aim of the thesis is to demonstrate an approach towards bottleneck detection based on real-time data for a flexible manufacturing system. It involves developing a template/tool in Microsoft Excel by identifying important parameters from the data collected to identify the bottleneck machine.

1.5 Scope and Delimitation

The thesis focusses more on identifying important parameters and preparing a hypothesis/method on how machine learning algorithms can be implemented in the future to improve and automate the process of bottleneck identification using real-time data. The scope of the project is limited to developing a method to identify bottleneck for one of the flexible manufacturing cells only. Since the data collection method is similar in the other FMC's, the method can be easily implemented thereafter in other cells. The scope does not include how to improve the bottleneck machine but is limited to only identification. The machine signals such as (working, idle, breakdown, set-up) could not be recorded because of software issue. Due to this, a different approach to calculate the machine states had to be used by utilizing the PLC signals.

1.6 Product and Production system of the case company - SKF

The thesis has been conducted at the SKF AB plant in Gothenburg, Sweden and is focused on the manufacturing of spherical roller bearings.

1.6.1 The Company

SKF AB (Swedish - Svenska Kullagerfabriken, English - Swedish Ball Bearing factory) is the world's largest bearing manufacturer which was founded in Gothenburg, Sweden, 1907. The company manufactures and supplies bearings, seals, lubrication and lubrication systems, maintenance products, mechatronics products, power transmission products, condition monitoring systems and related services globally (Wikipedia, 2019).

SKF has made tremendous progress towards building new generation of manufacturing systems by embracing the power of digital technologies leading to improvement in speed, flexibility and efficiency (Nia Kihlström, 2017). SKF's spherical roller bearing production facility in Gothenburg serves as a test-bed for world-class digital manufacturing that includes a digital information network and advanced automated solutions such as robots and automated guided vehicles (AGV). SKF has initiated a 5G Enabled Manufacturing (5GEM) project by collaborating with Ericsson and Chalmers University of Technology to make the manufacturing processes and systems faster, cleaner, safer and more robust pioneering their way towards building a smart factory (Ericsson, 2017). The investments in these digital technologies has reduced the lead time, inventories and cost and have also helped them focus on their energy efficiency by reducing their energy usage by 16 percent in spite of a significant increase in the production output (Theo Kjellberg, 2015).

1.6.2 The Product and the Production System

The following figure 1.1 highlights the main parts of a Bearing.

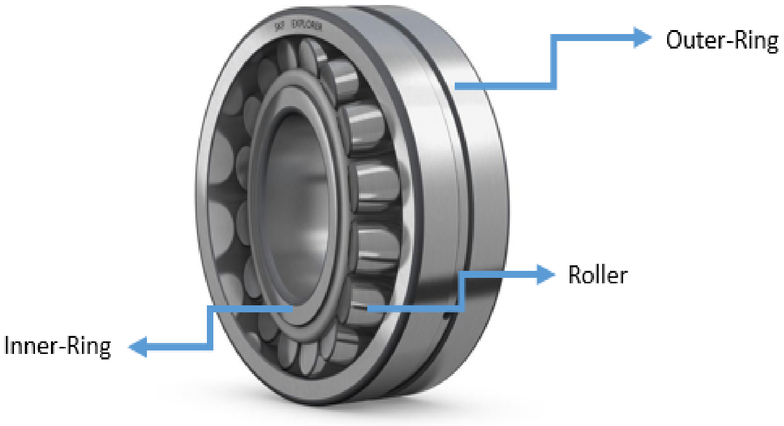


Figure 1.1: Spherical Roller Bearing

The thesis is focused on the production of inner-ring and outer-ring of a spherical roller bearing. The production system consisted of two Inner-ring FMC's, two Outer-ring FMC's, one Inner-ring Face-grinding FMC and one Outer-ring Face-grinding FMC. The Face-grinding FMC's has one robot while the Inner-ring and Outer-ring FMC's have two robots. The processes carried out in all the Inner-ring and Outer-ring FMC's are the same and in the Inner-ring and Outer-ring Face-grinding FMC's are the same. As the duration of the thesis was restricted to 20 weeks only, the Outer-ring FMC was selected for performing the thesis and the results would be replicated on other FMC's. The following figure 1.2 represents the layout of the various machines in the Outer-ring FMC.

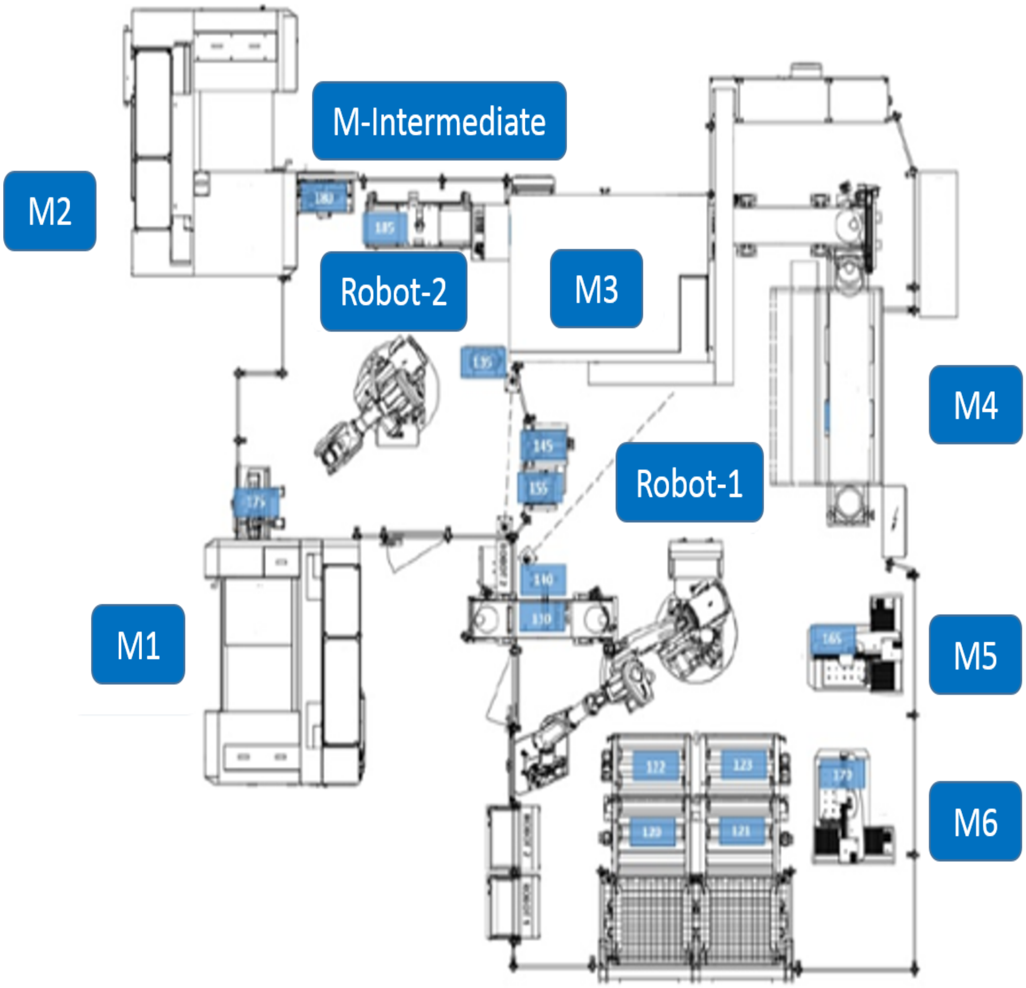


Figure 1.2: Flexible manufacturing cell - Layout

The Outer-ring FMC consists of two cells with a robot in each cell and a transfer station in between to transfer the part from one cell to another. The process flow being carried out in this FMC when a part is produced is as follows:

1. Robot-1 picks up the part from the input pallet.
2. Robot-1 places the part at the transfer station between Cell-1 and Cell-2.
3. Robot-2 picks up the part and places it in the inlet of machine M1.
4. Robot 2 picks up the part from the outlet of machine M1 and places it in the inlet of machine M2.
5. Robot-2 picks up the part from the machine M2 and places it in the Orientation platform before putting it in the machine M-Intermediate.

6. The part flows from the machine M-Intermediate into the machine M3.
7. The part flows from the machine M3 via a conveyor into the machine M4.
8. Robot-1 picks up the part from the machine M4 outlet and places it in either of the two machines M5 or M6.
9. Robot-1 picks up the part from either of the two machines M5 or M6 and places it on the outlet pallet.

The product flow is thus sequential and follows a line concept. The robot feed parts to machines as per requirements and signals received. Apart from these operations the Robot-1 performs some additional tasks such as picking and placing the paper on the pallet and sending the part for Quality inspection after machines M5 or M6 (Every 25th part varying as per batch).

The machines M1, M2, M-Intermediate, M3, M5 and M6 are discrete machines meaning that at a particular time only one product can be processed while the machine M4 is a continuous process meaning that at a particular time more than one product can be processed. The total number of products i.e. buffer quantity that each machine can hold is shown in Table 1.1:

Machine	Inlet	Slot	Process	Outlet	Total
M1	1	1	1	1	4
M2	1	1	1	1	4
M-Intermediate	-	-	1	-	1
M3	-	1	1	1	3
M4	1	-	6	1	8
M5	-	-	1	-	1
M6	-	-	1	-	1

Table 1.1: Total buffer quantities of the machines

2

Literature Study

This chapter will provide an overview of the research carried out within the area of bottlenecks in the manufacturing industry. It will also provide the reader the important knowledge on the different methods of bottleneck identification.

2.1 Bottlenecks in a Production system

Since the thesis focusses on identifying a suitable bottleneck detection method for a flexible manufacturing system, let us first understand the meaning of a bottleneck. A product has to go through a series of machines and undergo numerous operations on them before it is complete and is sent to the final customer. Along this journey, some activities are value-adding while the others are disruptions such as breakdown, set-up, lack of material, lack of operator, etc. It is these disruptions on a particular machine that result in blocking or starving of the upstream or downstream process in the production system. The machine which gets disrupted the most is the bottleneck as it disturbs the flow of material to the other machines and affects the production system performance (Subramaniyan, 2015).

There exists several bottleneck definitions in literature. A bottleneck is the machine or station that constrains the throughput of products produced in the production system (Betterton and Silver, 2012). Chiang et al., (2001) indicates that the reason behind a machine being a bottleneck is either because the machine is working too slow when it is producing or it is not available for production because of disturbances.

In order to stay competitive in the manufacturing industry, the production system must be dynamic in nature and should be able to produce different variants. The bottlenecks in such systems shift from one machine to another due to change in the machine parameters required for the different variants. This is termed as shifting bottlenecks (Roser et al., 2003).

Since bottleneck limits the throughput of a system, any improvement made on a machine other than the bottleneck will only contribute to more waiting time (Ericson, 2017). Thus, it is crucial to identify the bottleneck accurately so that the resources can be allocated to the machine efficiently in order to elevate the production system (Johannesson and Shams, 2018).

2.2 Bottleneck detection methods

Bottlenecks in a production system can be found in three different ways namely; analytical, simulation and data-driven (Ericson, 2017; Subramaniyan et al., 2016).

The analytical method utilizes a mathematical approach towards finding the bottleneck by calculating the overall capacity and comparing it with the throughput generated or by measuring the starving and blocking percentages from the measured production data (Ericson, 2017). However calculating these parameters is a time-consuming activity and involves approximations which makes it a difficult solution to implement in a complex production system (Subramaniyan et al., 2016).

Simulation based method comprises of a digital model of a production system. Such a model can give detailed information about utilization times, statistics of failures and maintenance activities and allows us to make changes and see how it affects the production system (Ericson, 2017). This can thus help in identifying bottlenecks in a complex production system (Li et al., 2009). The major drawback of using a simulation method is that it is time-consuming to develop the model and update the model as per the changes made in the production system. Also, the assumptions and misinterpretations can yield a different result since a simulation can never be as accurate to the real system (Ericson, 2017; Li et al., 2009).

Data-driven bottleneck detection method performs analysis on the data gathered from the Manufacturing system (Ericson, 2017). Manufacturing companies are generating and storing immense amount of data using sensor technologies. A study at an automotive company in Sweden revealed that 500,000 data rows are collected per year per machine (Subramaniyan et al., 2016). Such massive amount of big data generated provides opportunities to make more fact-based decisions by performing advanced analytics on real-time data which is generally not possible with the human-mind (O'Donovan et al., 2015).

Following are a few bottleneck detection methods that come under the above mentioned categories.

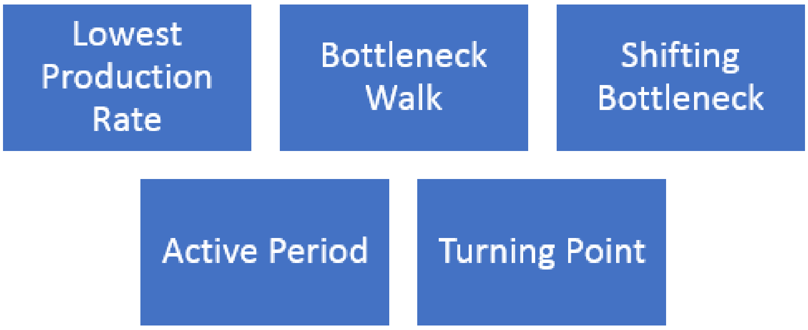


Figure 2.1: Bottleneck detection methods

2.2.1 Lowest production rate

Many industries have been finding the bottleneck today by simply looking at the cycle times and the production rates of the machines. Since the machine with the longest cycle time would have the lowest capacity and thus be the bottleneck (Kuo et al., 1996). Goldratt and Cox (1993) successfully proved that a well-balanced system with the same cycle times for all the machines in reality would still never produce one part per cycle time due to random disturbances such as breakdowns or human errors. Another way is by using a takt diagram as mentioned by Singh et al., (2011), where the process cycle times are measured and compared with each other and the customer takt time. The machine with a takt time greater than the customer takt time is the current bottleneck.

2.2.2 Bottleneck walk

A method that is not dependent upon data or theoretical models but rather focused directly on the shop floor is called as the bottleneck walk. The method is based on the starving and blocking concepts by observing the buffer quantities in the system since the buffers would tend to be full upstream and empty downstream when seen from the bottleneck (Roser et al., 2015). Thus, the bottleneck can be seen without performing any calculations and the reason could also be found out by mere observation. Roser et al., (2015) proposed this method and recommended not to automate the process of reporting the buffer levels as there is more value in manual observation.

2.2.3 Active period

An active period is defined as the period of time when the machine is either producing a part i.e. working, down due to breakdown or failure or when a machine is going through a set-up change for producing a different product type. The time period when the machine is idle or waiting because of another machine is not included in the calculations and termed as inactive period (Subramaniyan et al., 2018). The method utilizes the machine states such as working, breakdown, idle and set-up to compute the percentage of time the machine is active during a particular period of production time and thereby identifying the potential bottleneck (Subramaniyan et al., 2018). The method can be implemented on different types of production system by taking into account the actual times and does not require the timestamps when the machine had its active period (Ericson, 2017). Figure 2.2 is an example of how the machine states change from producing to downtime and to part-changing i.e. active and inactive states over a period of time.

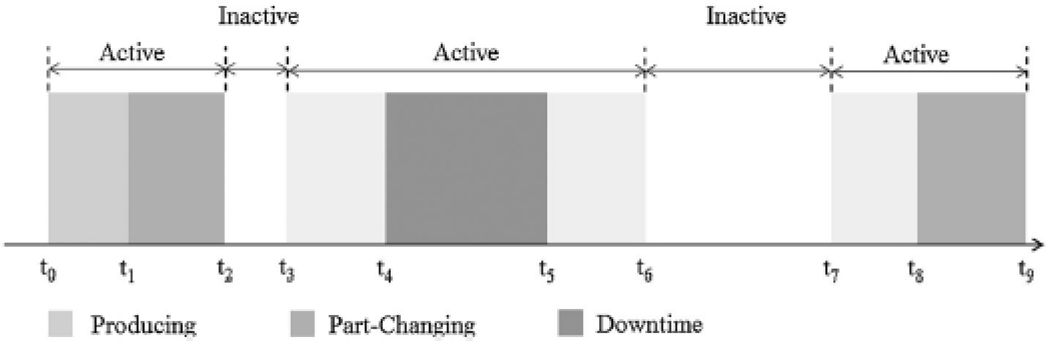


Figure 2.2: Active and Inactive states of the machine
Source: Adopted from Roser et al. (2002)

2.2.4 Shifting bottleneck

The shifting bottleneck method is based on when a machine is active or inactive. An active state is one where the machine is either producing, or being interrupted by a breakdown or set-up. An inactive state is when the machine is idle or waiting for another machine i.e. being blocked or starved (Ericson, 2017; Roser et al., 2002; Roser et al., 2003). At any given time, the machine with the longest active period is the bottleneck. The longer a machine is active, the more likely it is to limit the system's performance by starving or blocking other machines in the production system (Roser et al., 2002). When the machines with the longest active period overlap, the bottleneck is said to be shifting between these two machines. The total time a machine is sole and shifting bottleneck is summarized over the period of time to get the bottleneck machine (Ericson, 2017; Roser et al., 2003). The method can be implemented on complex systems by collecting data on the active periods and the timestamps for the active periods from the manufacturing execution system (MES) providing us results on a real-time basis (Ericson, 2017). Figure 2.3 shows how the bottleneck shifts from one machine to another.

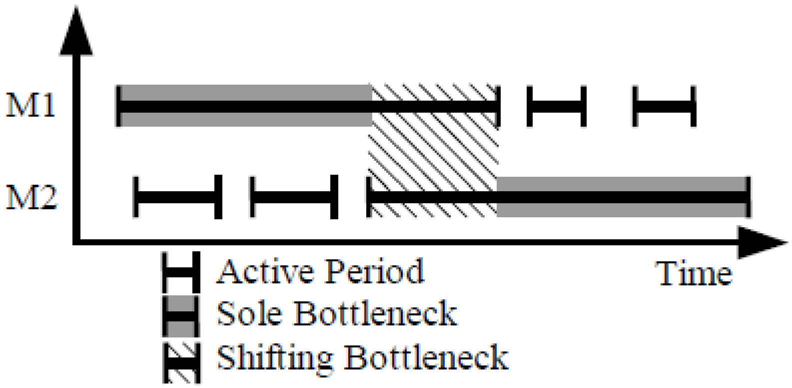


Figure 2.3: Shifting Bottlenecks
Source: Adopted from Roser et al. (2003)

2.2.5 Turning Point

The turning point method proposed by Li et al., 2009 utilizes online measurable production data to calculate blocking and starving states of the machine along with the respective buffer levels to identify the constraints without building any analytical or simulation model (Li et al., 2009). The nature of the material flow on the manufacturing system is reflected in the blocking/starving pattern. The bottleneck machine causes a variation in blockage and starvation. A bottleneck machine will cause the upstream machines to be blocked and the downstream machines to be starved (Li et al., 2009; Betterton and Silver, 2012). Also, a bottleneck machine will have a lower total of blocking plus starving time. Thus, a turning point is the station or machine where the trend of blockage and starvation changes from blockage being higher than starvation to starvation being higher than blockage. This indicates that the turning point has the highest percentage of operating and downtime compared to other machines in the system (Betterton and Silver, 2012). The proposed method can provide accurate results for short-term bottleneck detection and the opportunity to make performance improvement measures at any point of time. The method fails to detect true bottlenecks within a system consisting of large buffers and frequent small stoppages (Li et al., 2009). The Fig 2.4 indicates that Machine 4 has the smallest sum of blockage plus starvation time which makes it the turning point and the potential bottleneck.

Figure 2.4: Trend of blockage and starvation times in a production line
Source: Adopted from Li et al. (2009)

2.3 Flexible Manufacturing System

As per the current market situation, the demand and specifications from a customer changes very rapidly and the manufacturing system must be able to adapt to these changes as quickly as possible (El-Tamimi et al., 2012; Jovanovic, 2015). A flexible manufacturing system (FMS) is a highly integrated manufacturing system capable of quick configurations to produce a variety of products. An FMS generally consist of CNC machine tools, robots for material handling, automated storage and retrieval system and computers. A typical FMS can process a number of different product types without human intervention which makes it possible to produce a variety a parts in less time and cost (El-Tamimi et al., 2012). For a manufacturing system to be flexible, Jovanovic (2015) has identified the following criteria:

1. Ability to identify and distinguish different product varieties.
2. Quick changeover of operating instructions.
3. Quick changeover of physical set-up.

3

Methodology

In this chapter, detailed overview of the methodology used in carrying out the thesis work is presented. The aim of the chapter is to give the reader overview of how the problem was analyzed.

The flowchart in figure 3.1 gives an overall approach to problem solving.

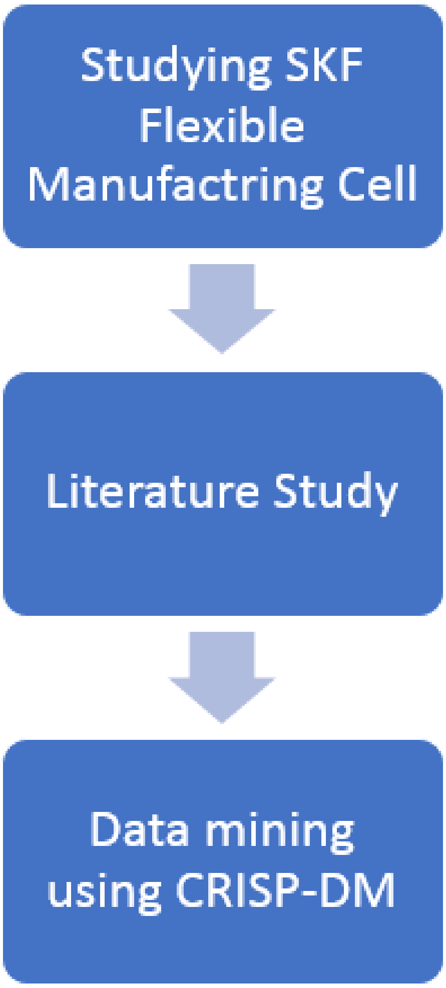


Figure 3.1: Approach to problem solving

The product and the production system is explained in detail in Chapter 1 and literature study relevant to the concepts used in Chapter 2.

The entire thesis work highly involves data analytics, so the Cross-industry standard reference for data mining (CRISP-DM) methodology was followed for data mining. The CRISP-DM methodology can be visualized in the figure 3.2

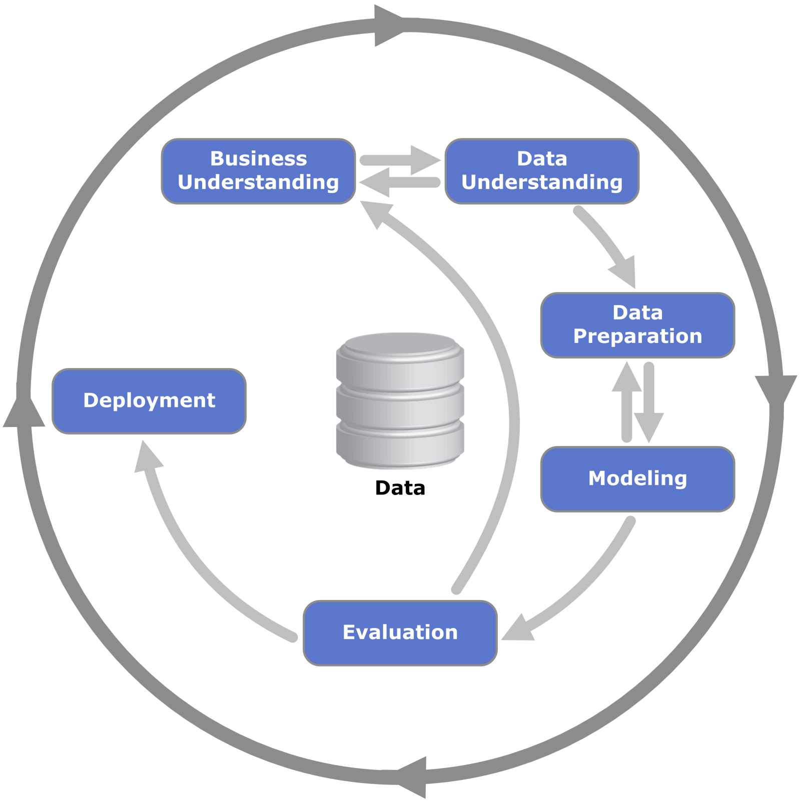


Figure 3.2: CRISP-DM Process Diagram (adopted from Wikipedia)

3.1 Business Understanding

The first step of the project was to have an understanding of the business objective of the primary stakeholder, SKF AB and align it with the work to be carried out in the thesis. The primary objective of SKF AB was to identify real-time shifting bottleneck in their world class spherical roller bearing flexible manufacturing cell and develop an analytic tool to identify the same. Identifying real-time bottleneck would enable SKF AB to make better informed and fact-based decisions to work on the bottleneck machine and enhance their throughput and productivity.

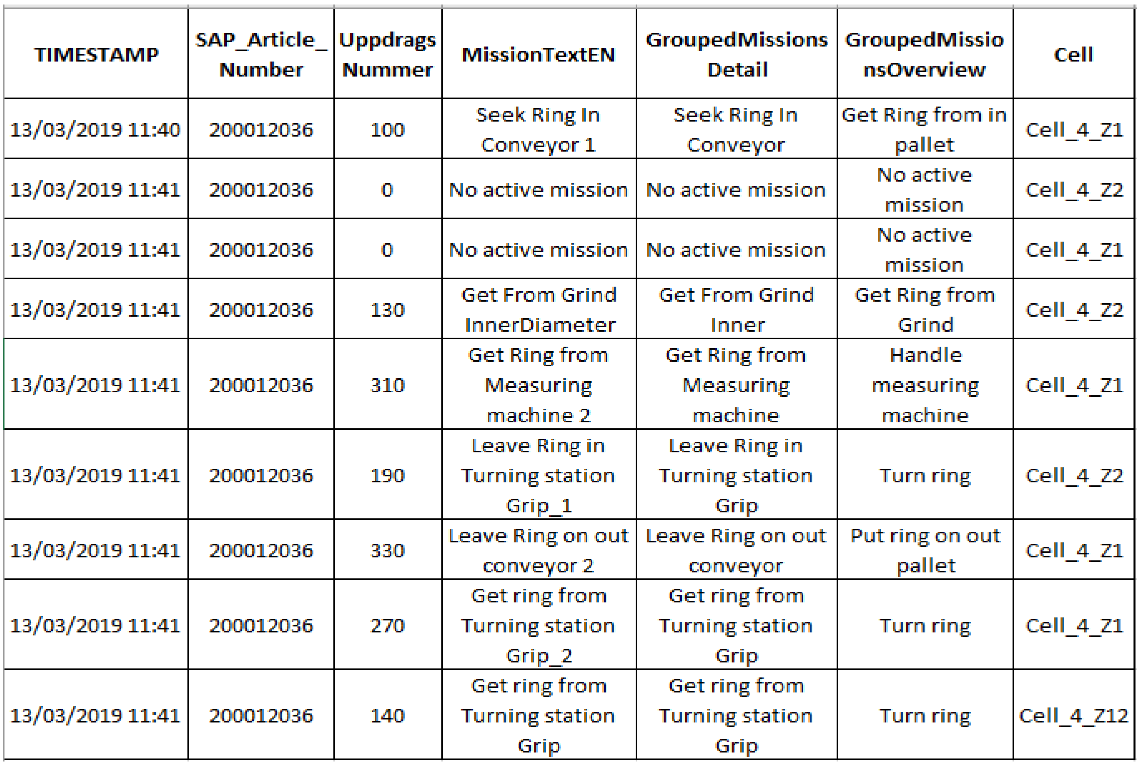
3.2 Data understanding

Data understanding step involves getting access to data, exploring the data using visualization methods in the form of tables and charts and determining the overall

quality of the data (IBM Corporation 1994, 2012)
Data understanding in CRISP-DM methodology consists of 4 steps which are explained below in subsection 3.2.1 to 3.2.4.

3.2.1 Collection of initial data

In this step, the data sources from which data is acquired along with their location, methods used to acquire the data and problem encountered are recorded (Smart Vision Europe, 2018).
The initial data set was imported from SQL server database at SKF AB and imported into MS Excel for processing and analysis. Table 3.1 gives an overview of the data-set imported from SQL.



TIMESTAMP	SAP_Article_Number	Uppdrags Nummer	MissionTextEN	GroupedMissionsDetail	GroupedMissionsOverview	Cell
13/03/2019 11:40	200012036	100	Seek Ring In Conveyor 1	Seek Ring In Conveyor	Get Ring from in pallet	Cell_4_Z1
13/03/2019 11:41	200012036	0	No active mission	No active mission	No active mission	Cell_4_Z2
13/03/2019 11:41	200012036	0	No active mission	No active mission	No active mission	Cell_4_Z1
13/03/2019 11:41	200012036	130	Get From Grind InnerDiameter	Get From Grind Inner	Get Ring from Grind	Cell_4_Z2
13/03/2019 11:41	200012036	310	Get Ring from Measuring machine 2	Get Ring from Measuring machine	Handle measuring machine	Cell_4_Z1
13/03/2019 11:41	200012036	190	Leave Ring in Turning station Grip_1	Leave Ring in Turning station Grip	Turn ring	Cell_4_Z2
13/03/2019 11:41	200012036	330	Leave Ring on out conveyor 2	Leave Ring on out conveyor	Put ring on out pallet	Cell_4_Z1
13/03/2019 11:41	200012036	270	Get ring from Turning station Grip_2	Get ring from Turning station Grip	Turn ring	Cell_4_Z1
13/03/2019 11:41	200012036	140	Get ring from Turning station Grip	Get ring from Turning station Grip	Turn ring	Cell_4_Z12

Table 3.1: Example of data-set imported from SQL

3.2.2 Data Description

In this step, the format, quantity, field identities and surface features of the data acquired are explained (Smart Vision Europe, 2018)
The imported data-set consisted of 7 columns and each column description is given below:

- Timestamp: Date and time when the mission is performed.
- SAP_Article_Number: Currently running product type in the cell.
- Uppdrags Nummer: It depicts the predefined number given to actions performed by the robot. '0' mission number depicts that robot is not performing any action i.e. robot is in idle state.

- Mission Text EN: Description of the action performed by the robot.
- Grouped Mission Detail: It gives generalized information already defined in the Mission Text EN column.
- Grouped Mission Overview: Actions i.e. robot missions are combined to show one complete robot activity.
- Cell: Cell number of whose data is imported from SQL. Z1 and Z2 represent the two robots in the flexible manufacturing cell Outer-Ring Cell.

3.2.3 Data Exploration

Data exploration step is used to address data mining questions. Querying, visualization and reporting techniques are used to explore the data. Initial findings and their impact on the project is reported (Smart Vision Europe, 2018)

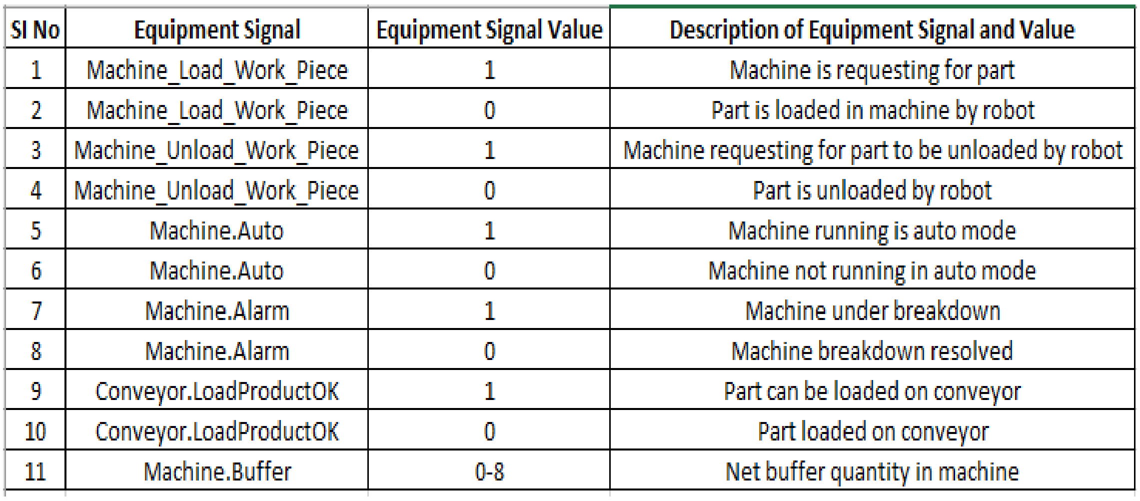
Data exploration was done using visualization based on how the data is reported. Data visualization led us to conclude that the imported data set could not be used for any of the real-time bottleneck detection methods. The imported data-set can give us information of the robot working and idle states and not of the machines in the cell.

3.2.4 Verification of data quality

The quality of data is examined based on the completeness, correctness and errors in the data-set. If quality is ascertained to be poor, possible suggestions to improve the quality are reported (Smart Vision Europe, 2018).

Overall data quality was ascertained to be poor with regards to the business objective of identifying real-time bottleneck. SQL data gave us insights about the actions performed by the robot and information regarding machine states (working, idle, breakdown and set-up) could not be derived for analytic purpose.

Since machine states were not being recorded, no real-time bottleneck detection method except for bottleneck walk and lowest production rate could be used. To be able to calculate machine states, a different approach by utilizing PLC signals i.e. communication between robot and machines in the cell is proposed. Keeping this approach in mind, robot missions and corresponding columns are excluded from the data-set and additional columns consisting of equipment signal, equipment signal value, product designation instead of SAP article number were recorded. Description of additional signals is given in Table 3.2



Sl No	Equipment Signal	Equipment Signal Value	Description of Equipment Signal and Value
1	Machine_Load_Work_Piece	1	Machine is requesting for part
2	Machine_Load_Work_Piece	0	Part is loaded in machine by robot
3	Machine_Unload_Work_Piece	1	Machine requesting for part to be unloaded by robot
4	Machine_Unload_Work_Piece	0	Part is unloaded by robot
5	Machine.Auto	1	Machine running is auto mode
6	Machine.Auto	0	Machine not running in auto mode
7	Machine.Alarm	1	Machine under breakdown
8	Machine.Alarm	0	Machine breakdown resolved
9	Conveyor.LoadProductOK	1	Part can be loaded on conveyor
10	Conveyor.LoadProductOK	0	Part loaded on conveyor
11	Machine.Buffer	0-8	Net buffer quantity in machine

Table 3.2: Description of Equipment Signal and Equipment Signal Value

3.3 Data preparation

Data preparation in CRISP-DM consists of steps explained below in subsection 3.3.1 to 3.3.4.

3.3.1 Data Selection

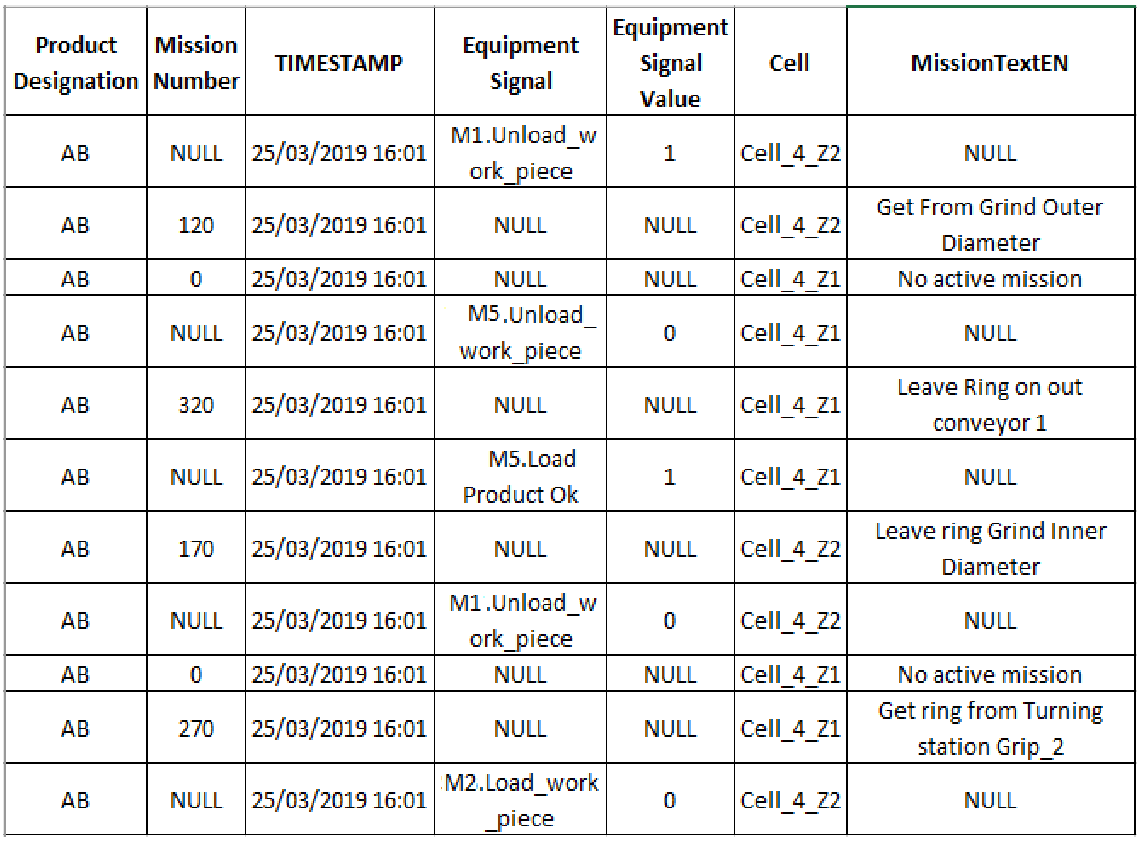
In data selection step, the data to be included/excluded is listed and reasons for data inclusion and exclusion is mentioned (Smart Vision Europe, 2018).
As the data quality imported from SQL was poor, additional signals as described in Table 3.2 were recorded. Table 3.3 gives an overview of SQL data-set after additional signals were incorporated.

3.3.2 Data cleaning

Data cleaning is performed to improve the data quality to an acceptable level which is suitable for analysis. It can involve selection of certain subsets of data and data estimation by modeling techniques (Smart Vision Europe, 2018).
Data quality was further improved keeping in mind the business objective. The quality was found to be acceptable except for duplication of rows in case of buffer signals. Analytic methods are used to clean data.
Finally, for data analysis below mentioned columns were used.

- Product Designation
- Timestamp
- Equipment Signal
- Equipment Signal Value
- Cell

It was ensured that during data cleaning, the data is free from signal errors such as duplication of signals.



Product Designation	Mission Number	TIMESTAMP	Equipment Signal	Equipment Signal Value	Cell	MissionTextEN
AB	NULL	25/03/2019 16:01	M1.Unload_work_piece	1	Cell_4_Z2	NULL
AB	120	25/03/2019 16:01	NULL	NULL	Cell_4_Z2	Get From Grind Outer Diameter
AB	0	25/03/2019 16:01	NULL	NULL	Cell_4_Z1	No active mission
AB	NULL	25/03/2019 16:01	M5.Unload_work_piece	0	Cell_4_Z1	NULL
AB	320	25/03/2019 16:01	NULL	NULL	Cell_4_Z1	Leave Ring on out conveyor 1
AB	NULL	25/03/2019 16:01	M5.Load Product Ok	1	Cell_4_Z1	NULL
AB	170	25/03/2019 16:01	NULL	NULL	Cell_4_Z2	Leave ring Grind Inner Diameter
AB	NULL	25/03/2019 16:01	M1.Unload_work_piece	0	Cell_4_Z2	NULL
AB	0	25/03/2019 16:01	NULL	NULL	Cell_4_Z1	No active mission
AB	270	25/03/2019 16:01	NULL	NULL	Cell_4_Z1	Get ring from Turning station Grip_2
AB	NULL	25/03/2019 16:01	M2.Load_work_piece	0	Cell_4_Z2	NULL

Table 3.3: Example of data-set imported from SQL after incorporating additional signals

3.3.3 Construct Data

Data construction step involves building of derived attributes or generation of new records or the transformed values of existing attributes (Smart Vision Europe, 2018)

The aim was to include working times, non working times, breakdown times, daily production time (start and end of production), signal error (to incorporate time included in analysis because of multiple signals being generated).

Data was constructed keeping in mind the conditions derived for machine to be in blocked state or starved state.

Data imported from SQL into excel was filtered to contain machine specific signals that would be required to calculate blocking and starving time. The results were analyzed using pivot tables and charts.

Additionally following times were calculated:

- Stop time: Its the non working time of the machine when its not under breakdown.
- Set up time
- Breakdown time
- Processing/Working time

3.3.4 Data Integration

In data integration, information from multiple databases, tables or records is combined (Smart Vision Europe, 2018).
Data integration was limited to aggregation of results (machine states time) for each machine in the flexible manufacturing cell and analysis purpose.

3.4 Modeling

Modeling stage consists of selecting a suitable modeling tool/technique, generation of test design, model building and assessment (Smart Vision Europe, 2018).
The data is modeled in MS Excel using in-built excel functions and conditions derived which are explained in results table 5.2.

3.5 Evaluation and Validation

In the evaluation step, the results obtained are evaluated using the business criteria established in the business understanding step (IBM Corporation 2004, 2012)
The model was developed using a week's data-set and validated using another week's data-set. Model is evaluated on the basis of whether it fulfills the business objective i.e. identifying the real-time bottleneck.

3.6 Deployment

Deployment step consists of two activities, namely, result monitoring and project review (IBM Corporation 1994, 2012)
However, in this thesis work, deployment step was excluded as the proposed algorithm was not implemented physically on the FMC.

4

Results

In this chapter, the results of the thesis work are presented. This section also comprises of the following:

1. Starving and Blocking conditions of the machines in the flexible manufacturing cell.
2. Pictorial representation of starved and blocked states of the machines using machine signals.
3. Flowchart depicting the diagrammatic representation of the proposed algorithm to calculate the machine states time (Blocking, Starving, Stop, Breakdown, Set-up).

4.1 Machine Conditions

The most important aspect of the thesis work after data cleaning was to draft the blocking and starving conditions for the machines. Table 5.2 gives an overview of the starving and blocking conditions of each machine in the cell. Machine signals (Load(0), Load(1), Unload(0), Unload(1)) timestamps are available in the imported data-set. For calculating blocking time, processing time is also required. The brief explanation of calculating processing time in the conditions table 5.2 is described below:

Calculation of processing time: Since processing time is not explicitly available in the imported data-set for each cycle, a different approach is used for calculating it. At every instance when the net buffer after unloading becomes '0', the time from Load(0) i.e. (Part loaded at the input) to Unload(1) i.e. (Part ready for unloading at the output) is calculated for each product type. This condition of buffer '0' is chosen because we can arrive at the actual cycle time as it would be the first part entering and leaving the machine. The calculated time also comprises of the travel time in the machine which must be excluded from the processing time calculations. For our analysis, we chose to take into account the minimum cycle time from the bunch of values measured at buffer '0' condition because it would be closest to the ideal cycle time as machine will not be in blocked or starved state. Travel times are mentioned in table 4.2 for each machine.

$$\boldsymbol{ProcessingTime = (Unload(1) - Load(0)) - TravelTime}$$

Machine	**Starving Condition**	**What to measure? Time from-**	**Blocking Condition**	**What to measure? Time from-**
M1	Net buffer must be ’0’ after unloading.	Unload(1) till Load(0)	Net buffer in M1>’1’ at Unload(1) and M2 signal is Load(0).	Unload(1) till Unload(0) - Processing Time
M2	Net buffer must be ’0’ after unloading.	Unload(1) till Load(0)	Net buffer in M2>’1’ at Unload(1) and M-Intermediate signal is Load(0).	Unload(1) till Unload(0) - Processing Time
M3	Net buffer must be ’0’ after unloading.	Unload(1) till Load(0)	Net buffer in M3>’1’ at Unload(1) and Conveyor signal is Load(0).	Unload(1) till Unload(0) - Processing Time
M4	Net buffer must be <’6’ after unloading.	Load(1) till Load(0)	Net buffer in M4 at Unload(1) is ’7’ or ’8’ and signal at M5 and M6 is Load(0).	Unload(1) till Unload(0)
M5	Single piece flow. Machine will starve for the time it requires part till it gets a part.	Load(1) till Load(0)	Machine is blocked for the time it sends a signal to unload till part is unloaded.	Unload(1) till Unload(0)
M6	Single piece flow. Machine will starve for the time it requires part till it gets a part.	Load(1) till Load(0)	Machine is blocked for the time it sends a signal to unload till part is unloaded.	Unload(1) till Unload(0)

Table 4.1: Starving and Blocking conditions for the machines

Machine	**Travel time for calculating Processing time**
M1	Time from inlet to slot (10 sec) + Changeover time (5 sec)
M2	Time from inlet to slot (10 sec) + Changeover time (5 sec)
M3	Part transferred via conveyor. Pre-processing and post-processing travel time = 35 sec
M4	Not applicable
M5	Single piece flow. Travel time not required.
M6	Single piece flow. Travel time not required.

Table 4.2: Component travel time (used for calculating processing time)

4.2 Starving and Blocking conditions for each machine

A pictorial representation has been used to demonstrate the above mentioned conditions defined for calculating the starving and blocking times of the individual machines.
Note: The following points must be kept in mind as they will be used quite often.

1. I = Input position (The robot loads the part here)
2. S = Slot position (The part moves on to this slot after it has been loaded at the input and waits to be processed)
3. P = Processing position (The part is processed here)
4. O = Output position (After the part is processed, the part comes to the output where it is picked up by the robot for the next operation)
5. While calculating the starving and blocking times for all the machines, it is ensured that both the timestamps are for the same product-type and have taken place on the same date since set-up time and daily production time have been calculated separately.
6. The defined machine states are also checked while measuring the starving and blocking times. It is ensured that the machine is in working condition and remains in working condition during the period where the starving and blocking time is measured as breakdown time and stop time have been calculated separately.
7. All the formulas for starving and blocking use the PLC signals of the machine that is under consideration.

4.2.1 Starving condition

The starving condition for each machine can be explained as follows:

4.2.1.1 Machine (M1), Machine (M2) and Machine (M3)

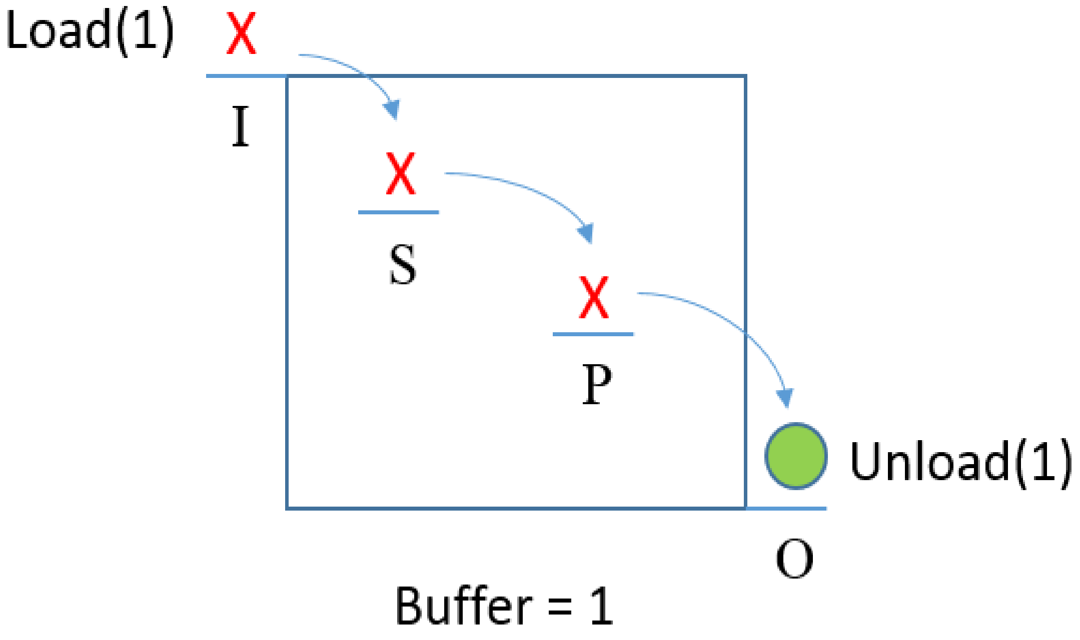


Figure 4.1: Machine (M1) and Machine (M2) Starving condition

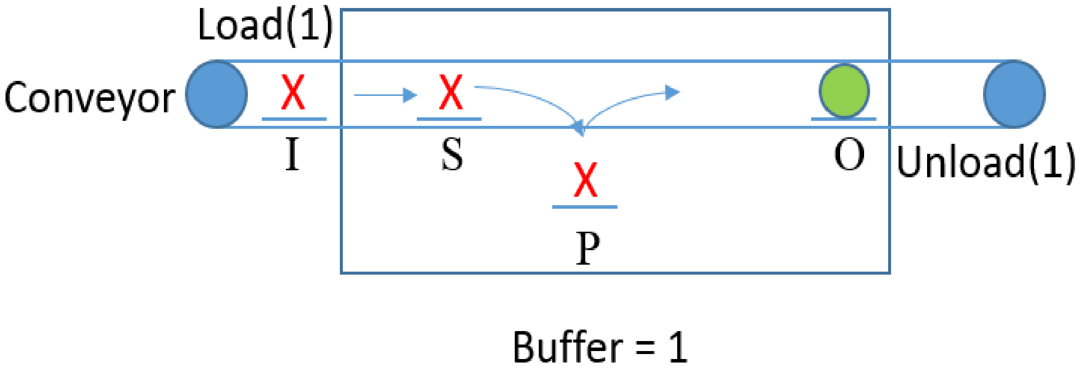


Figure 4.2: Machine (M3) Starving condition

The starving condition for the machines M1, M2 and M3 are the same since they all have a buffer capacity of four and a similar product flow. A machine is said to starve when the net buffer in the machine is zero. Consider the machine layout in Fig 4.1 and Fig 4.2 where the machine has one part at the outlet indicating that the processed part is ready to be picked by giving the Unload(1) signal and has a net buffer of one. The machine has no part in any other positions. The machine has a Load(1) signal at the input indicating it needs a part. Once the part at the outlet is picked up by the robot, the net buffer in the machine will change to zero. However, the machine is not processing a part since the time it gave the Unload(1) signal. Thus, when the net buffer changes to zero, the time from when the machine gave it's last Unload(1) signal to the time the machine gives a Load(0) signal indicating that a part has been loaded successfully is defined as the starving time. The formula for the starving time for the following machines can be written as:

$$\boldsymbol{StarvingTime = Load(0) \text{ - } Unload(1)}$$

4.2.1.2 Machine (M4)

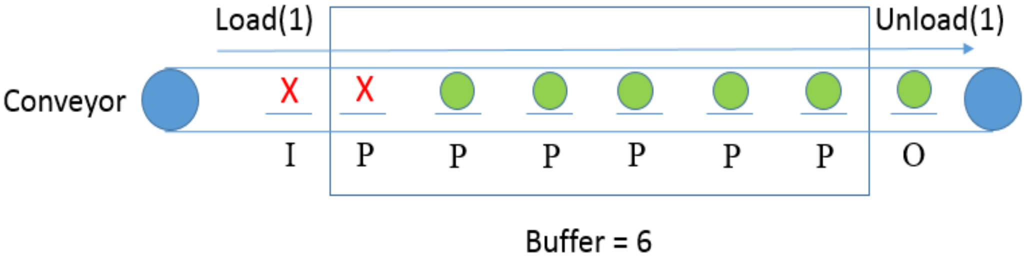


Figure 4.3: Machine (M4) Starving condition

The total buffer capacity of the machine M4 is eight parts with the possibility to process six parts at a time and one part each at the input and outlet. Consider the machine layout in Fig 4.3 where the net buffer is six with one part at the outlet. The M4 will starve if it has a net buffer of less than six parts after unloading of the part at the outlet since the machine can process a maximum of six parts at a time. Thus, after the part is picked up by the robot from the outlet, the buffer condition is checked if it is less than six and time from when it gave its last load(1) signal i.e. (Request for a part) till the time it gave its Load(0) signal i.e. (Part received) would give us the starving time. In the situation shown in Fig 4.3, the machine is starving for one part. The formula for calculating the starving time can be written as:

$$\boldsymbol{StarvingTime = Load(0) \text{ - } Load(1)}$$

4.2.1.3 Machines (M5 and M6)

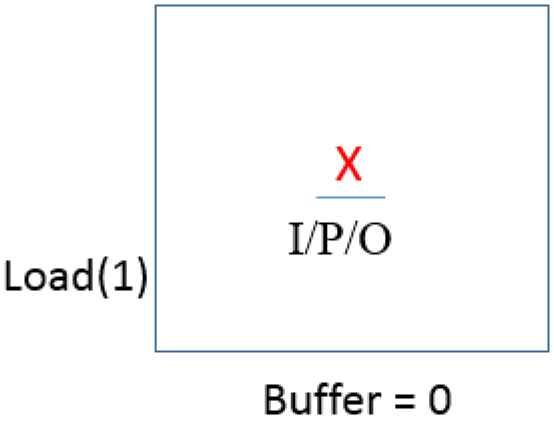


Figure 4.4: Machines (M5 and M6) Starving condition

The machine layout for the M5 and M6 can be seen in Fig 4.4. The machines M5 and M6 are exactly identical and are used for measuring purpose. Since it is a single piece flow with a capacity of one part, the starving time is calculated from the time the machine gives a Load(1) signal i.e. (Request for a part) till the time it gives a Load(0) signal i.e. (Part received). In the fig 4.4, the machine has no part and is therefore starving. The formula can be written as:

StarvingTime = Load(0) - Load(1)

4.2.2 Blocking condition

The blocking condition for each machine can be explained as follows:

4.2.2.1 Machine (M1)

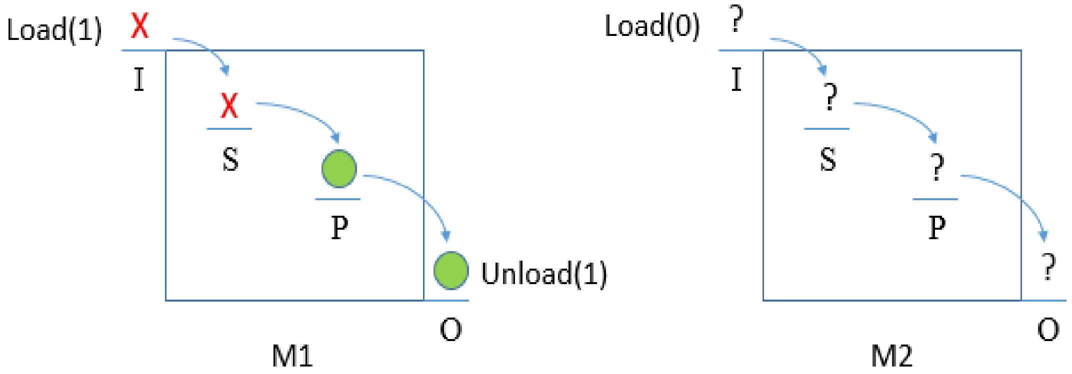


Figure 4.5: Machine (M1) Blocking condition

The machine M1 can be blocked at instances when it has a net buffer greater than one i.e. (2,3,4). Apart from this, M1 must have a signal Unload(1) i.e. (Part ready to unload) and M2 must have a signal Load(0) i.e. (Not ready to load a part) are two other conditions that must be satisfied . The net buffer in M2 at that moment is not important as long as it has a Load(0) signal. If above two conditions are satisfied, the blocking time is calculated from the time the M1 gave an Unload(1) signal i.e. (Part ready to unload) till the time it gives a Unload(0) signal i.e. (Part unloaded successfully) and the processing time is subtracted (Explained in Section 4.1). Consider one of the situations when the M1 will be blocked in Fig 4.5. In this case, it has a net buffer of two with a part ready to be unloaded but the M2 has a Load(0) signal and thus it cannot unload it. The "?" in the M2 indicates that the position of the part can vary and there exists many possibilities when M2 can give a Load(0) signal. The formula for the blocking time for the M1 can be written as:

$$\boldsymbol{BlockingTime = (Unload(0) - Unload(1)) - ProcessingTime}$$

4.2.2.2 Machine (M2)

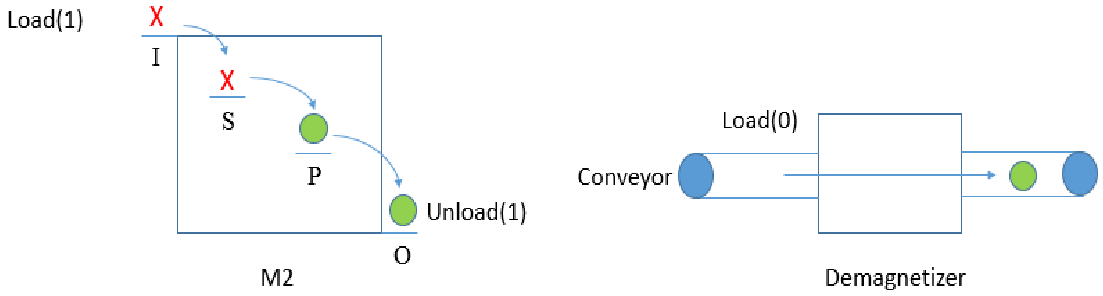


Figure 4.6: Machine (M2) Blocking condition

The blocking condition for the machine M2 is very much similar to M1. It can be blocked at instances when the net buffer is greater than one i.e. (2,3,4). Apart from this, the two conditions that must be satisfied are that M2 must have a signal Unload(1) and the machine M-Intermediate must have a signal Load(0). The M-Intermediate is present between the M2 and the M3, thus its signal is taken into account rather than M3. The M-Intermediate has a single piece flow and if it has a part either at its beginning or at the end it would not accept another part. Consider the situation in Fig 4.6, where the M-Intermediate cannot accept another part and has a Load(0) signal. The calculation for the blocking time remains the same if these conditions are satisfied. The formula for the blocking time for the M2 can be written as:

$$\boldsymbol{BlockingTime = (Unload(0) - Unload(1)) - ProcessingTime}$$

4.2.2.3 Machine (M3)

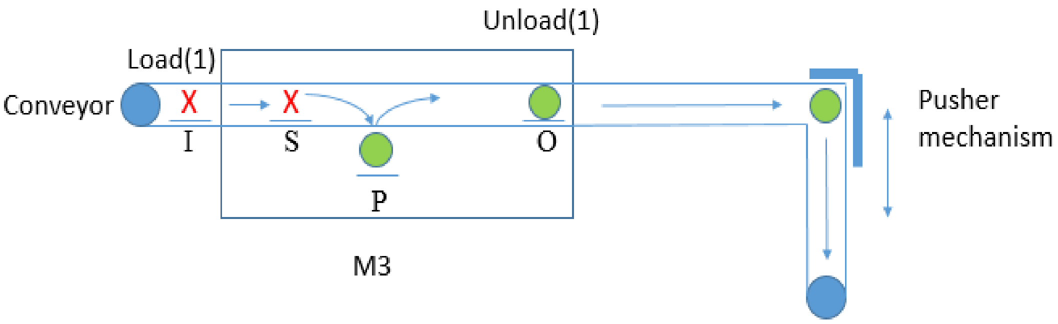


Figure 4.7: Machine (M3) Blocking condition

The blocking condition for the machine M3 is similar to M2 and can be blocked at instances when the buffer is greater than one i.e. (2,3,4). Apart from this, the conditions that must be satisfied for the M3 to be blocked are that the M3 must have a signal Unload(1) and the conveyor system must have a signal Load(0). The conveyor system is present between the M3 and M4 and since it has a single piece flow, it plays a key role in the blocking of the M3. Consider the situation in Fig 4.7, where the M3 has a part ready to be unloaded but it cannot do so since the conveyor has not transported the part to the M4. There is a pusher mechanism in the conveyor which gets activated when the part reaches the pusher and then it is sent to the M4. The conveyor will give a Load(1) signal i.e. (Ready to load a part) once the pusher mechanism is back to its initial position. The calculation for the blocking time remains the same if these conditions are satisfied. The formula for the blocking time for the M3 can be written as:

$$\textit{BlockingTime} = (\textit{Unload(0)} - \textit{Unload(1)}) - \textit{ProcessingTime}$$

4.2.2.4 Machine (M4)

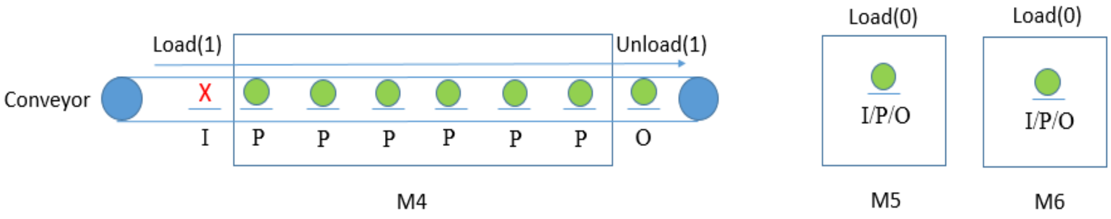


Figure 4.8: Machine (M4) Blocking condition

The blocking condition for M4 is a little different from the other machines. The M4 can be blocked when the net buffer is either seven or eight with a part waiting to be unloaded at the output. Additionally, machines M5 and M6 must have a signal Load(0) respectively. As seen in Fig 4.8, the current capacity in M4 is seven with a part ready to be unloaded. However, the part cannot be unloaded as the machines M5 and M6 already have a part. In this machine, as the process is continuous and six parts can be under processing at the same time, the processing time cannot be subtracted because it is difficult to measure it with the data available. The formula for the blocking time for the M4 can be written as:

$$\textit{BlockingTime} = (\textit{Unload(0)} - \textit{Unload(1)})$$

4.2.2.5 Machines (M5 and M6)

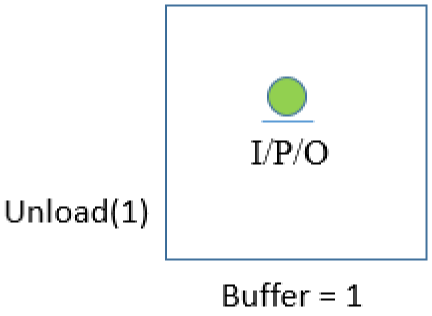


Figure 4.9: Machines (M5 and M6) Blocking condition

As the machines M5 and M6 have single piece flow, the blocking time is calculated from the time the machine gives a Unload(1) signal i.e. (Part ready to unload) till the time it gives a Unload(0) signal i.e. (Part unloaded successfully). In the fig 4.9, the machine has a part and is therefore blocked. The formula can be written as:

$$\textit{BlockingTime} = (\textit{Unload(0)} - \textit{Unload(1)})$$

4.3 Algorithm

The proposed algorithm for calculating the machine states time is a result of CRISP-DM methodology explained in Chapter 3 and carefully drafted starved and blocked conditions of the machines explained in section 4.2.

The algorithm is specifically meant for calculating the machine states time. After machine states time has been calculated, different bottleneck detection methods can be used to identify the bottleneck.

The two algorithms in Figure 4.10 and Figure 4.11 corresponds to calculating the stop time, breakdown time, set-up time and machine waiting times (starving and blocking) for machine M1. The algorithm is the same for different machines in the cell except for different buffer quantities which can be referred to from Table 1.1

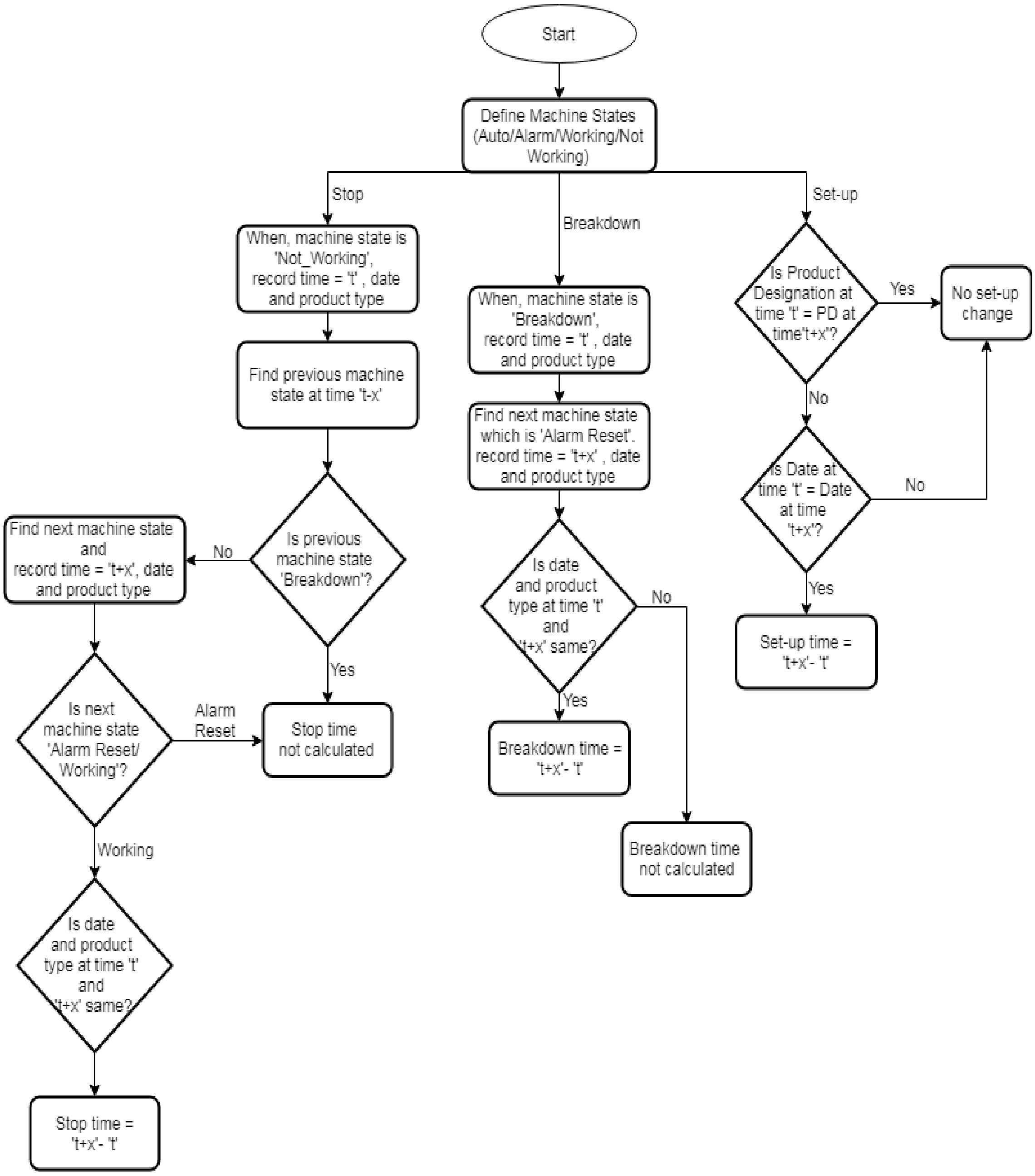


Figure 4.10: Flowchart depicting algorithm for calculating breakdown, stop and set-up time

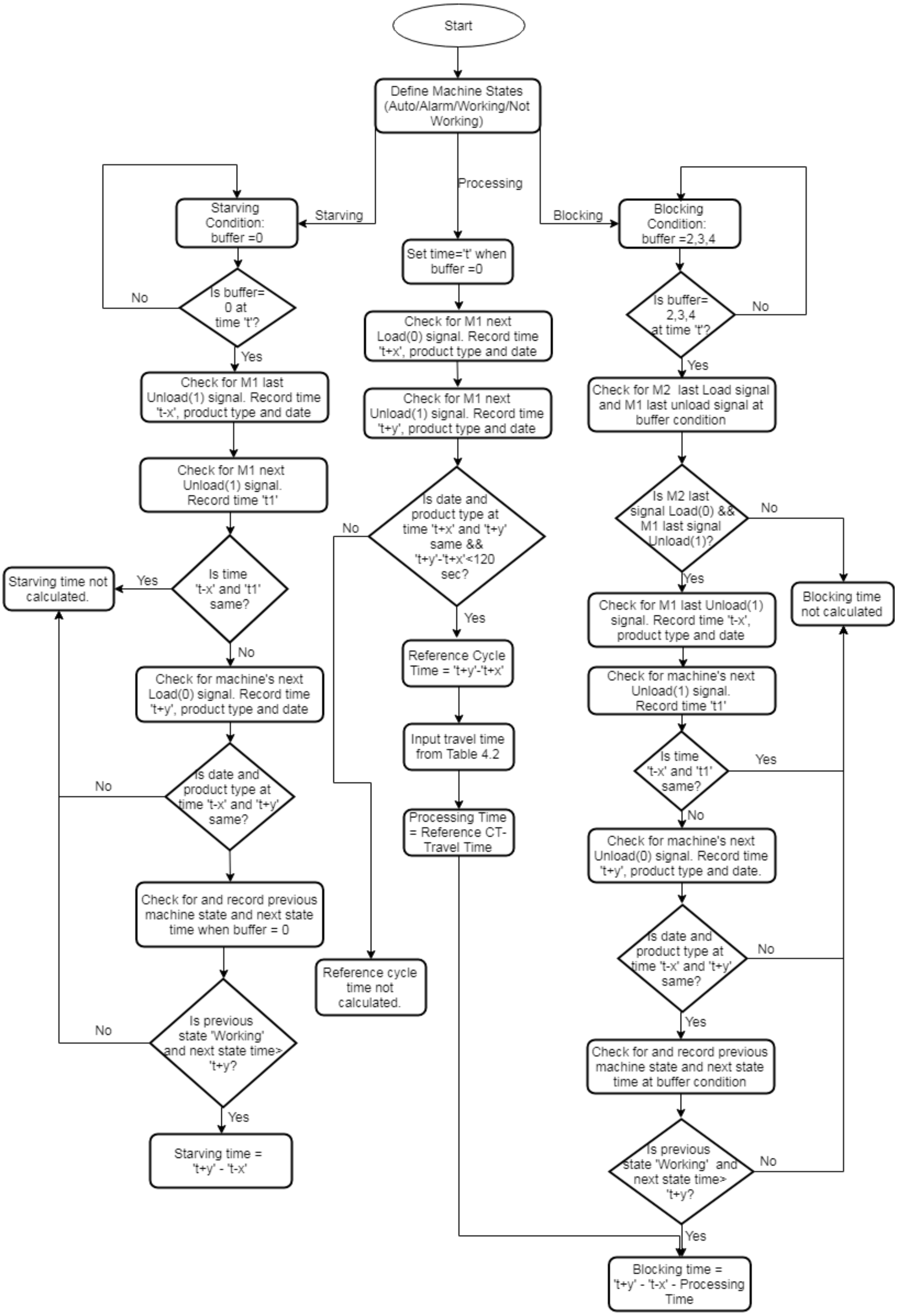


Figure 4.11: Flowchart depicting algorithm for calculating starving, blocking and processing time

4.4 Bottleneck detection using Data Analytics

The turning-point method of bottleneck identification was tested and data has been analyzed as follows:

4.4.1 Day-wise analysis

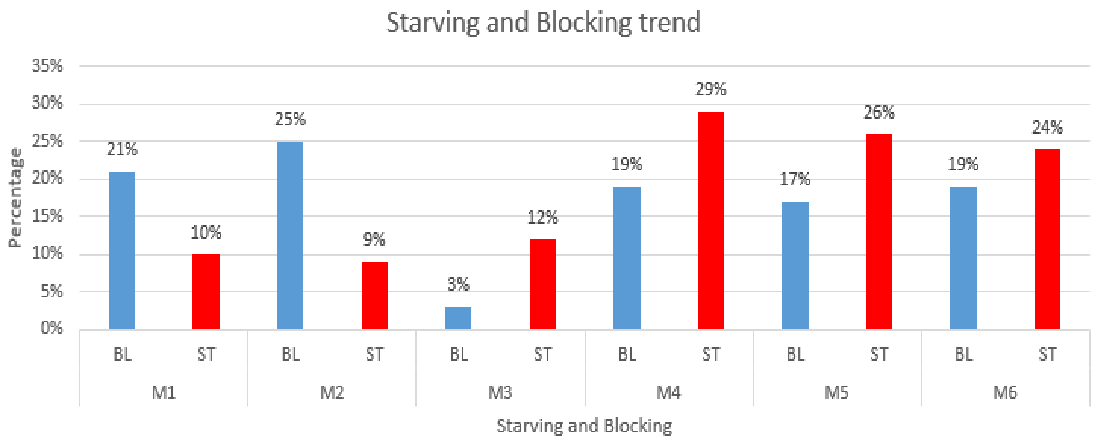


Figure 4.12: Starving and blocking rates for the different machines

The starving and blocking rates were calculated to identify the bottleneck by analyzing the production data for one day. As can be seen in Fig 4.12, based on the literature study conducted, the turning point happens at M3 and the trend changes from blockage being higher than starving to starving being higher than blocking. As M3 has the smallest sum of starving and blocking, it is the bottleneck for this particular day.

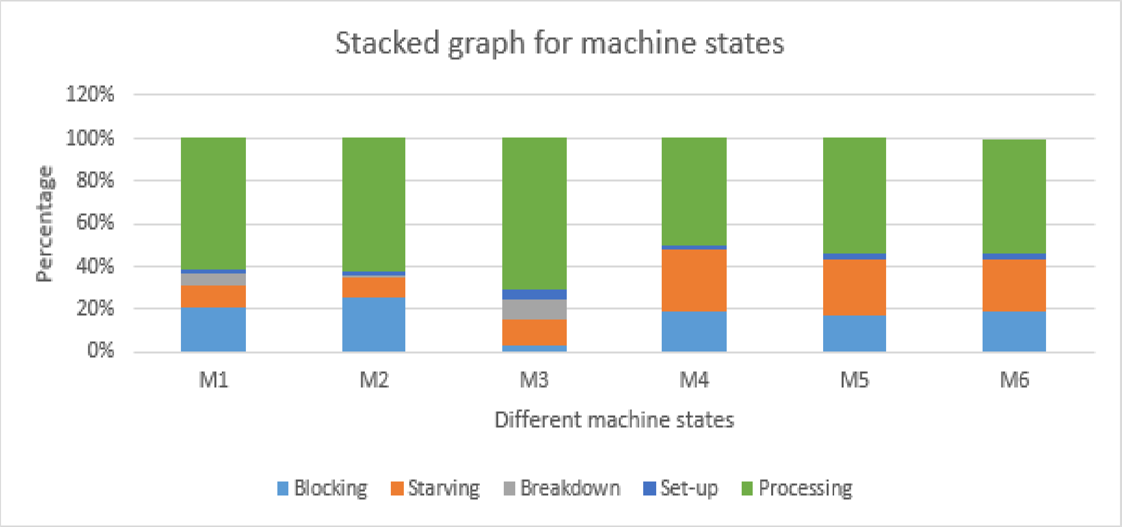


Figure 4.13: Categorization of the machine states

The Fig 4.13 shows the stacked percentages of the different machine states for a particular day in production (Same day in consideration with Fig 4.12). The active period (Breakdown + Set-up + Processing) is highest for M3 which indicates that it also has the highest active period and hence is the bottleneck.

4.4.2 Week-wise analysis

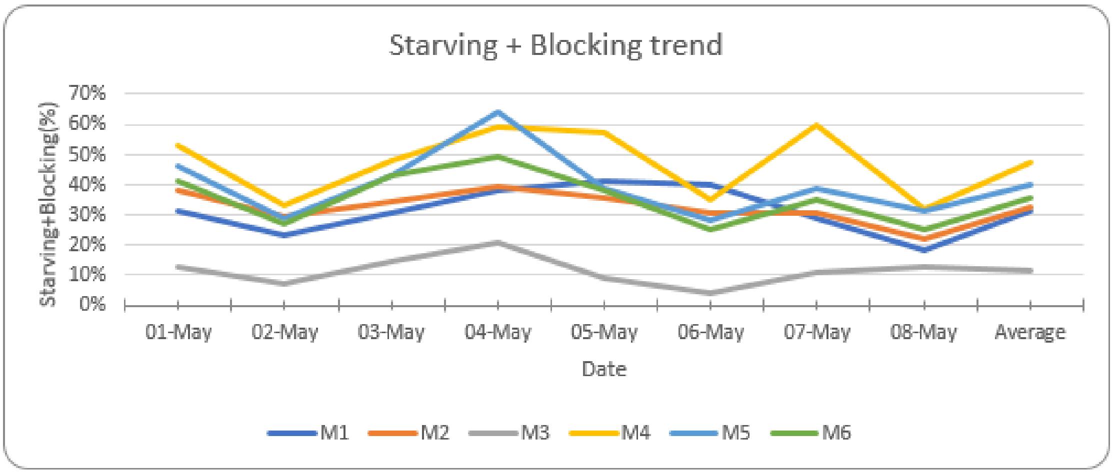


Figure 4.14: Starving and blocking rates for the different machines

The Fig 4.14 shows the starving plus blocking trend for the different machines for one week. M3 has the smallest percentage of starving plus blocking everyday. However, there is a definite variation in the rates of the other machines every day. Thus, the M3 is the bottleneck as per the turning point bottleneck detection method.

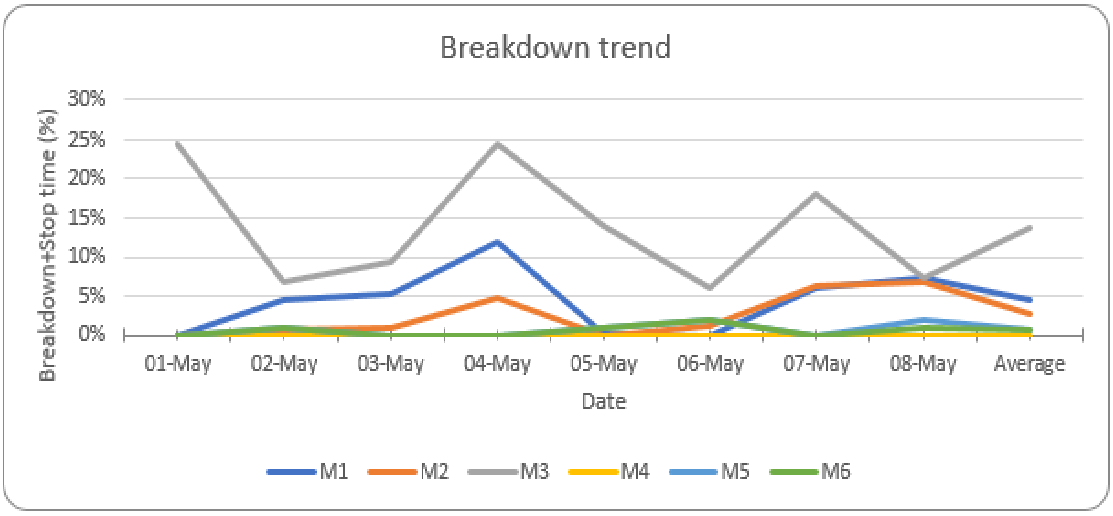


Figure 4.15: Breakdown rates for the different machines

As can be seen in the Fig 4.15, The breakdown rates (breakdown time + stop time) are highest for the M3 everyday for the entire week. Thus, adding to the high active period of M3. M4 breakdown data is not being recorded. M5 and M6 have almost close to 0 percent breakdown on an average.

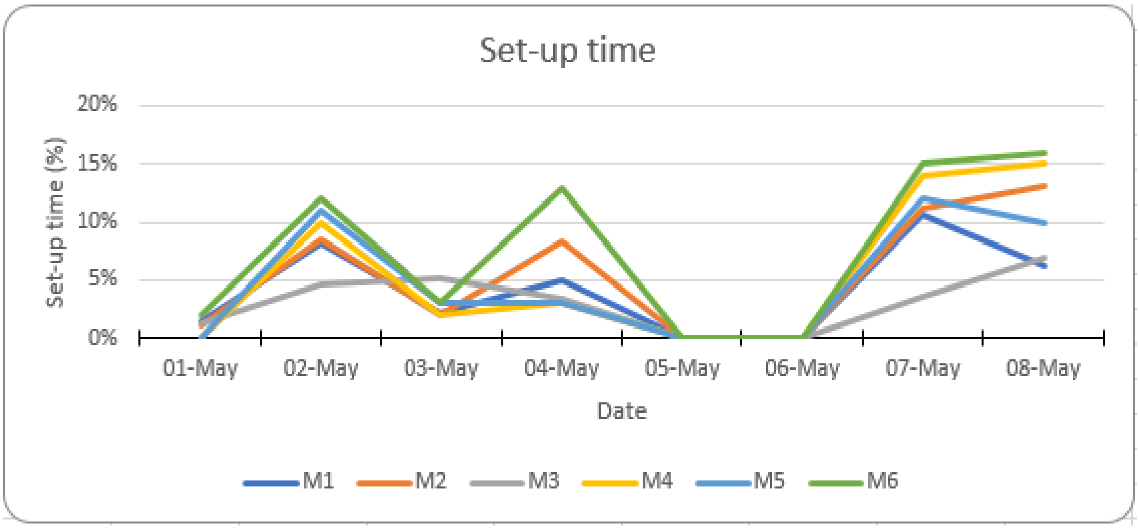


Figure 4.16: Set up time for different machines

As can be observed from Fig 4.16, set up constitute 0 to 15 percent. The machines with the highest contribution to set up is M4, M5 and M6 but the analysis is misleading since only program change is required for set up change on these machines. Set up time reflected for M4, M5 and M6 is actually the machine waiting for part after set up change in the cell.

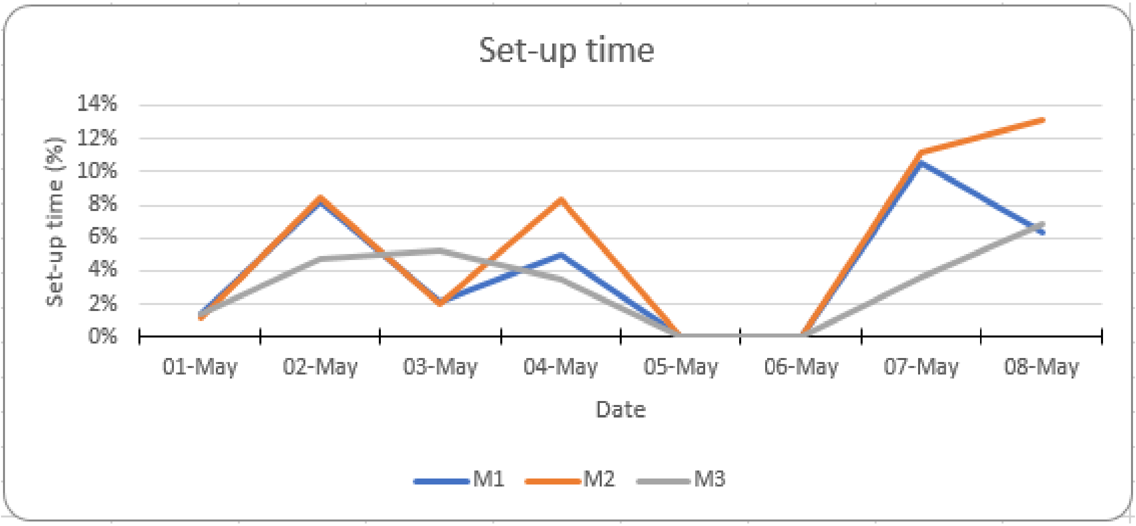


Figure 4.17: Set up contribution of different machines in total production time

Above Fig 4.17 shows the set up time for M1, M2 and M3. Set up time contribution is highest for M2 machine. Analysis for set up time for different product type can also be done with the algorithm defined in section 4.3.

4.5 Comparison with Lowest Production Rate and Active Period bottleneck detection methods

The identified bottleneck machine M3 using the turning point will be compared with the bottleneck machine identified with the 2 bottleneck detection methods, namely, lowest production rate and active period.

4.5.1 Lowest Production Rate

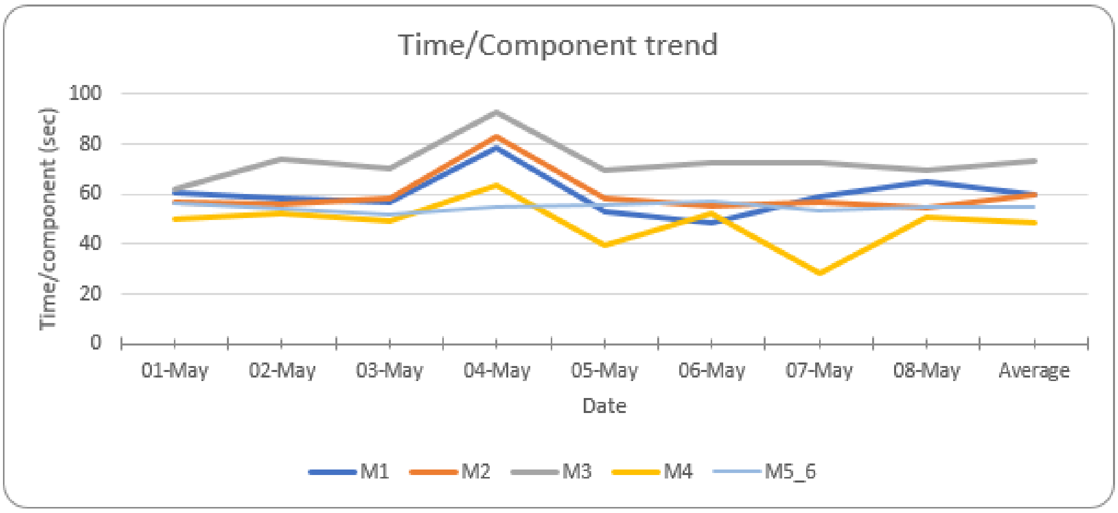


Figure 4.18: Time/component rates for the different machines

The Fig 4.18 shows the time/component each of these machine take everyday during the entire week. Since the M5 and M6 are in parallel, the average of both of them have been taken for this calculation. The machine M3 clearly takes more time than the rest of the machines to produce a part and hence is the bottleneck machine based on lowest production rate bottleneck detection method. On an average, M1 and M2 take nearly the same amount of time.

4.5.2 Active Period

The active period of a machine comprises of the working time, breakdown time and set-up time. The machine with the highest active period percentage is the bottleneck. As can be observed in Fig 4.19, machine M3 has the highest active period each day and hence the bottleneck is machine M3.

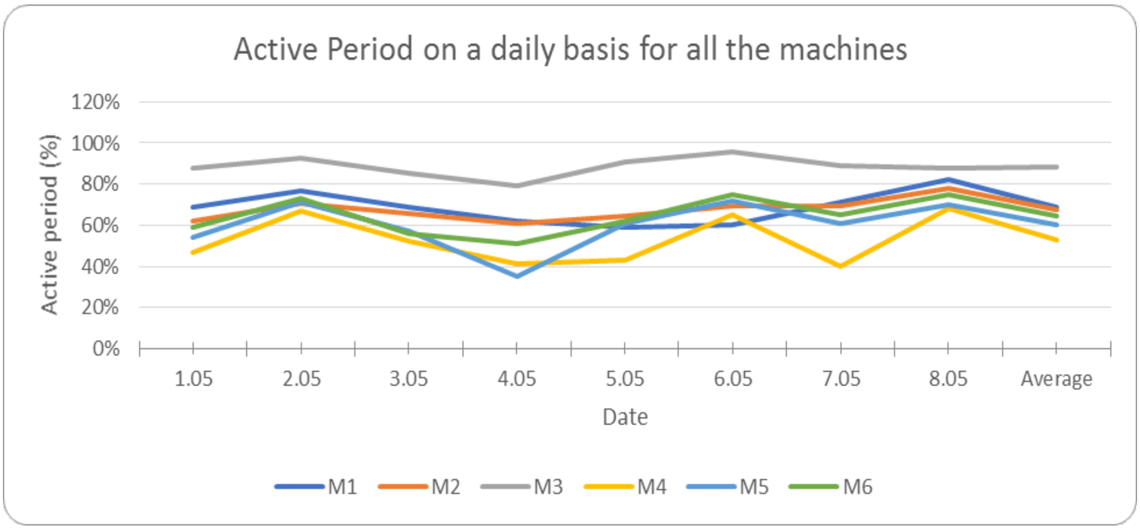


Figure 4.19: Active Period comparison for the different machines

5

Discussion

The following chapter will discuss the methods and results and discuss how this study can be useful in the future.

5.1 Methods and Result

The aim of the thesis was to develop an analytic tool to demonstrate an approach towards identifying bottleneck in a flexible manufacturing system. A literature study was conducted on the bottleneck detection methods that are based on real-time data analytics. Most of the studies were conducted on serial production lines and not on a flexible manufacturing system. However, in this thesis a flexible manufacturing cell was modelled as a serial line for implementation of a suitable bottleneck detection method. Previous studies have utilized Active-period method due to the availability of machine states making it suitable for the implementation of data-driven algorithms (Subramaniyan et al., 2016).

The data gathered and collected by SKF was mined using CRISP-DM methodology. An alternative approach utilizing machine PLC signals to calculate machine states time is demonstrated and an algorithm is proposed for the same. The proposed algorithm is generic in nature and can be used as a guideline to approach similar problems. One of the limitation of the proposed algorithm is the calculation of machine states which is specific to the data-set being recorded at SKF AB. It can be implemented across all FMC's that manufacture spherical roller bearings at SKF AB since the data structure would be identical to the data being recorded in the test cell. Another limitation of the algorithm is in the calculation of set-up time. Set-up time calculated using the algorithm constitutes the waiting time for the first part to arrive at the machine for processing.

Turning Point method is used when the production system does not have large buffer quantities and less frequent minor stoppages and can thus detect the true short-term bottleneck (Li et al., 2009). The choice of using Turning Point method for bottleneck identification for the production system under consideration stems from the fact that with the data-set available it was easier to calculate blocking and starving rates more accurately. Another reason to use Turning Point method was the lack of sensor data which could monitor or help calculate the processing time of the part accurately. Processing time was calculated after calculating waiting, breakdown and set-up time and then subtracting it from the total production time. Calculation of processing time resulted in the possibility of using active period method for bottleneck identification.

For the Turning Point method, the starving and blocking rates for all the machines in the test cell were calculated. A template has been created in MS Excel for each machine that calculates their respective starving and blocking rates. Separate files had to be made since the data was massive and a lot of formulas have been used to make the results as accurate as possible. Data cleaning and preparation consumed a lot of time as additional parameters had to be measured and only two weeks of data could be gathered for the purpose of analysis. After analyzing the final data-set, machine M3 was identified as the bottleneck as it had the lowest sum of starving plus blockage times indicating the turning point happening at M3 (Betterton and Silver, 2012). Also, the active period of M3 was the highest since it had the maximum amount of time in processing and breakdown. Due to this, the M3 also had the highest time for producing a part compared to the other machines in the cell. The identified bottleneck using Turning Point method i.e. M3 was also the bottleneck machine identified by the lowest production rate and active period percentage bottleneck detection methods (Subramaniyan et al., 2018; Kuo et al., 1996). Two weeks data showed M3 as the bottleneck machine and no significant variation was observed. However by improving the data quality and analysing the trend over a longer period of time might suggest different bottleneck machine on a short-term basis. The proposed method also provides the opportunity to identify the bottleneck machine based on the production of a particular product type. Thus, identifying the true bottleneck is important to improve the throughput of the system because any improvements made on any other machine would only increase waiting time (Ericson, 2017).

Bottleneck Detection Method	**Requirement**
Turning Point Method	Starving and Blocking rates
Active Period Percentage	Machine States (Working, Breakdown, Set-up)
Shifting Bottleneck	Active Period
Lowest Production Rate	Cycle Time
Bottleneck Walk	Manual inspection of buffer quantities

Table 5.1: Requirements for the different Bottleneck detection methods

Table 5.1 shows the specific requirements for the use of different real-time bottleneck detection methods. The proposed algorithm for calculating machine states has been developed based on the data-set available at SKF AB. The calculated machine states (working, breakdown, set-up, idle) can then serve as a platform to detect bottleneck using the bottleneck detection methods mentioned in table 5.1.

Points of comparison	**Analytic tool to identify bottleneck in a flexible manufacturing cell**	**Data driven shifting bottleneck detection algorithm**
Machine States	Calculated using machine PLC signals. Machine states defined as Working, Not working, breakdown and alarm reset.	Machine states are prerequisite for implementing the algorithm
Time	Signals measured at discrete time interval (not constant)	Signal measured at constant discrete time interval (10 sec)
Method	Turning Point	Active Period with shifting bottleneck
Algorithm	Developed conditions in the data-sets to calculate machine states time using timestamps	To identify shifting bottleneck using Active Period
Bottleneck Type	Average	Current, Average and Non-bottleneck
Requirement	Pre-defined machine PLC signals, time stamps and product type	Machine state information and timestamps
Calculation	Blocking and starving time, Previous, current and next machine state, Production time, Breakdown time, stop time and set up time	Active state and Inactive state
Production System	Flexible manufacturing cell	Serial production line

Table 5.2: Comparison between proposed bottleneck detection method and data-driven shifting bottleneck detection algorithm (Subramaniyan et al., 2016)

Table 5.2 shows a comparative study between the proposed method in this thesis to identify the bottleneck with the data-driven shifting bottleneck detection algorithm (Subramaniyan et al., 2016). One interesting conclusion is that the proposed algorithm in this thesis work to calculate machine states time can be extended to use data-driven shifting bottleneck detection algorithm.

5.2 Future scope

While this project has been conducted and tested on only one of the flexible manufacturing cells at SKF, the method can be verified on other cells as well since the cell layout and machines used are the same. The current method requires manual input of data in excel and then performing analysis. Also, there is the issue of handling massive data with excel. To overcome this, the algorithm mentioned in Fig 4.10 and Fig 4.11 can be implemented and programmed so that input of data is automatic and the machine states time can be calculated on a real-time basis since the current method uses historical data.

The algorithm can give more accurate results if the quality of the data being recorded is improved. Data quality can be improved by the use of additional sensors from which data can be recorded to accurately calculate the processing time, breakdown time, set-up time. For instance, additional sensor can be placed on the Machine (M3) which gives a signal that part processing has started and when it has ended. This was also discussed with SKF AB and the proposed idea was accepted but due to resource constraints at SKF AB it was not implemented. The quality of data can be further improved by removing repetitive signals that are being measured currently. Since the current results are based on two weeks of accurate data, they must also be verified by performing this analysis on data for a month.

The thesis work is more focused on how to calculate machine states using PLC signals. Once states are calculated, different bottleneck detection methods could be used. Its not possible to identify reasons for specific states. For example, if machine is waiting or under breakdown, we cannot find the reasons for the same. Further exploration of different bottleneck detection methods can be made such as active period and shifting bottleneck. Robots are also a part of flexible manufacturing cell. Its possible to identify total waiting time of robot but its not possible to divide it in starving and blocking states. Hence, its excluded from our analysis.

6

Conclusion

In this chapter, the conclusions drawn from the work carried out are presented below:

- Aim of the thesis was fulfilled as a different approach towards bottleneck detection based on real-time data has been developed for a flexible manufacturing cell and bottleneck machine identified.

- Important parameters in the data-set has been identified and a template has been created for each particular machine within a FMC to calculate starving and blocking times.

- An alternative approach to calculate machine state time using PLC signals in an FMC is possible. The alternative approach can be used in case data-driven production management systems which monitor machine states are not installed.

- The bottleneck machine was successfully identified using Turning Point method and verified using lowest production rate and active period percentage bottleneck identification methods.

Bibliography

Betterton, C.E., Silver, S.J. (2012), "Detecting bottlenecks in serial production lines - A focus on interdeparture time variance", International Journal of Production Research, 50 (15), pp. 4158-4174.

Chiang, S.-Y., Kuo, C.-T., Meerkov, S.M. (2001), "c-Bottlenecks in Serial Production Lines: Identification and Application", Mathematical Problems in Engineering, 7 (6), pp. 543-578.

Wikipedia, (2012). "Cross-industry standard process for data mining", Retrieved from: https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining.

Drath, R. and Horch, A. (2014), Industrie 4.0: Hit or Hype? [Industry Forum]. IEEE Industrial Electronics Magazine, 8(2), pp.56-58.

El-Tamimi, A.M., Abidi, M.H., Mian, S.H., Aalam, J. (2012), "Analysis of performance measures of flexible manufacturing system", Journal of King Saud University - Engineering Sciences, 24 (2), pp. 115-129.

Ericson, A. (2017), "Master Thesis: Bottleneck detection in manufacturing", Chalmers Univesity of Technology.

Ericsson, (2017), “Manufacturing Intelligence”. Retrieved from https://www.ericsson.com/en/cases/2017/skf.

European Commission, (2016), “Big data and B2B digital platform”, Retrieved from: https://ec.europa.eu/growth/content/big-data-and-b2b-digital-platforms-next-frontier-europes-industry-and-enterprises-0_en.

Goldratt, E. Cox, J. (1993), "The Goal: A Process of Ongoing improvement".

Gölzer, P., Fritzsche, A. (2017), “Data-driven operations management: organisational implications of the digital transformation in industrial practice”, Production Planning and Control, Vol. 28, No. 16, pp. 1332-1343.

IBM Corporation (1994, 2012). "CRISP-DM Help Overview". Retrieved from: https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_overview.htm

Johannesson, A. Shams, P. (2018), "Master Thesis: Data-driven and Variant-based Throughput and Bottleneck Prediction Using Ensembled Machine Learning Algorithms", Chalmers University of Technology.

Jovanovic, S. (2015), "Flexible Manufacturing Systems and Quantitative Analysis of Flexible Manufacturing Systems", International Journal of Computer Applications, 132.

Kuo, C.-., Lim, J.-. Meerkov, S.M. (1996), "Bottlenecks in serial production lines: A system-theoretic approach", Mathematical Problems in Engineering, vol. 2, no. 3, pp. 233-276.

Liao, Y., Deschamps, F., Loures, E.F.R., Ramos, L.F.P. (2017), “Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal”, International Journal of Production Research, Vol. 55, No. 12, pp. 3609-3629.

Li, L., Chang, Q., Ni, J. (2009), "Data driven bottleneck detection of manufacturing systems", International Journal of Production Research, 47 (18), pp. 5019-5036.

Monostori, L., (2014), “Cyber-physical production systems: Roots, expectations and RD challenges”, Procedia CIRP, Vol. 17, pp. 9-13.

Nia Kihlström, (2017), "The SKF Smart Manufacturing Transformation". Retrieved from: https://www.skf.com/group/news-and-media/news-search/2017-06-20-the-skf-smart-manufacturing-transformation.html.

Neugebauer, R., Hippmann, S., Leis, M., Landherr, M. (2016), “Industrie 4.0 - From the Perspective of Applied Research”, Procedia CIRP, 57, pp. 2-7.

O’Donovan, P., Leahy, K., Bruton, K., O’Sullivan, D. T. J. (2015), "Big data in manufacturing: A systematic mapping study", Journal of Big Data, 1, 20.

Roser, C., Nakano, M., Tanaka, M. (2002), "Throughput sensitivity analysis using a single simulation", Winter Simulation Conference Proceedings, 2, pp. 1087-1094.

Roser, C., Nakano, M., Tanaka, M. (2003), "Comparison of bottleneck detection methods for AGV systems". Winter Simulation Conference Proceedings, 2, pp. 1192-1198.

Roser, C., Lorentzen, K. Deuse, J. (2015), "Reliable shop floor bottleneck detection for flow lines through process and inventory observations: the bottleneck walk", Logistics Research, vol. 8, no. 1, pp. 1-9.

Singh, B., Garg, S. K. Sharma, S. K. (2011), "Value stream mapping: literature review and implications for Indian industry", The International Journal of Advanced Manufacturing Technology, 53, 799–809.

Smart Vision Europe, (2018). "What is the CRISP-DM Methodology". Retrieved from: https://www.sv-europe.com/crisp-dm-methodology/

Subramaniyan, M., Skoogh, A., Salomonsson, H., Bangalore, P., Bokrantz, J. (2018), "A data-driven algorithm to predict throughput bottlenecks in a production system based on active periods of the machines", Computers and Industrial Engineering, 125, pp. 533-544.

Subramaniyan, M., Skoogh, A., Gopalakrishnan, M., Salomonsson, H., Hanna, A., Lämkull, D. (2016), “An algorithm for data-driven shifting bottleneck detection”, Cogent Engineering, Vol. 3, Issue 1, Article Number 1239516.

Subramaniyan, M. (2015), "Master Thesis: Production Data Analytics - To identify productivity potentials", Chalmers University of Technology.

Tayal, S., Nagwal, N., Sharma, K. (2018), “Role of big data in make in India”, Advances in Intelligent Systems and Computing, Vol. 564, pp. 431-437.

Theo Kjellberg, (2015). “SKF invests SEK 190 million in Gothenburg production channels”. Retrieved from: https://www.skf.com/group/investors/press-releases/skf-invests-sek-190-million-gothenburg-production-channels.

Wikipedia, (2019). “SKF”. Retrieved from: https://en.wikipedia.org/wiki/SKF.