



University of
Nottingham

UK | CHINA | MALAYSIA

Data driven methods for non-disruptive digitalisation of real manufacturing environments

Emil Tochev

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

October 2021

Abstract

As a result of current trends towards digital transformation and the integration of Industry 4.0 ideals and technologies, many manufacturers are racing to stay competitive in an increasingly digitised environment. Among them is Siemens Congleton, a manufacturer of variable speed drives and the funding body for this research project. This factory wants to maintain competitiveness in a changing manufacturing environment, but with minimal disruption to existing processes.

Initial assessments of the factory processes show it to use many modern techniques and technologies. Despite this, some areas of production still have scope to be improved. The first of these is the scheduling method in use: a Priority Rule Based Method (PRBM). An alternative has been proposed: a Mixed Integer Linear Programming (MILP) algorithm, capable of creating a schedule that accounts for the sequence dependant setup times present in this manufacturing process. This is tutorialised for non-technical readers and compared to a simplified version of the factory's scheduling method, on a simplified factory environment. Formulations of the algorithm are presented for single, serial and parallel setups, allowing for a modular approach to constructing a global scheduling method as well as local schedules for specific machine groups. In almost all ways the optimisation approach is better than or at least equal to the PRBM. The main drawback is the increased time taken and processing power required: however, this can be overcome by performing the calculations on a cloud platform and taking advantage of elastic computing technologies.

Another aspect of the Siemens factory that was identified as an opportunity to improve through digital transformation was the reactive maintenance and fault response procedures surrounding legacy machines. These machines cannot natively connect with wireless networks and create local data silos. As such, a generic Fault Detection and Health Monitoring (FD-HM) method using indirect data collection has been developed. It is presented in the context of a wave soldering machine use case, which suffers from solder build up in the pipes. A simple test bench is used to mimic the behaviour of the machine. Using an accelerometer and a Raspberry Pi, vibration data is collected. A gate valve creates the pipe blockage and a flow meter records the effect on the flow rate. This is then analysed using Bayesian Change point Detection (BCD) to detect and isolate the pump response when pipe blockages form. The results of the analysis are compared to the flow rate to assess the algorithm. The information can be used to identify the pump response in the future and inform machine users that maintenance is required. This reduces both the time necessary for establishing the presence of the machine fault, and the time spent using the machine while it is in that state.

This is followed by an online version of the BCD (OBCD) algorithm and an assessment of its effectiveness. By collecting incoming data points and

comparing them to the observed data already collected during the current cycle, the likelihood of a change point taking place can be assessed. This algorithm is intended to provide a real time indication of the machine state to operators.

Finally, the implementation and use of both the scheduling algorithm and FD-HM methods in a factory environment is discussed. A comparison between the MILP algorithm and the factory PRBM is presented with anonymised production times. An initial analysis of factory data is performed using the BCD and OBCD algorithms alongside the analysis of artificial data sets representing different potential machine fault scenarios. In addition to this, integration with Siemens' Industrial Internet of Things (IIoT) service, MindSphere, will be considered. To conclude, the application of a combined scheduling and FD-HM approach is presented and discussed.

Acknowledgements

I would like to thank my supervisors, Svetan Ratchev, Giovanna Martinez Arellano and Harald Pfifer, for their support throughout my PhD. I also want to thank Divish Rengasamy and Giovanna Martinez Arellano for their help with several key aspects of the project.

Many thanks as well to all who made me feel welcome and helped me gather information during my limited time at Siemens Congleton: Carl German, Anil Thomas, Prasad Manorathna, Tom Nowakowski, Humza Malik, Conner Bottomley, and Dean Cummins.

Finally, I would like to express my gratitude to my partner, friends, and family, for their unceasing belief and support.

List of Publications

1. E. Tochev, H. Pfifer, and S. Ratchev, ***“A Comparison of Centralised and Decentralised Scheduling Methods Using a Simple Benchmark System”***, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018 Bergamo, Italy, 11–13 June 2018; IFAC-PapersOnLine, vol. 51, no. 11, pp. 1287–1292, 2018.
2. E. Tochev, D. Rengasamy, H. Pfifer, S. Ratchev, ***“System condition monitoring through Bayesian change point detection using pump vibrations”***, 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), 2020, pp. 667-672
3. E. Tochev, H. Pfifer, S. Ratchev, ***“Indirect System Condition Monitoring Using Online Bayesian Changepoint Detection”***, IPAS 2020: Smart Technologies for Precision Assembly pp 81-92.
4. Y. Zhang, E. Tochev, S. Ratchev, C. German, ***“Production process optimization using feature selection methods”***, 13th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 17-19 July 2019, Gulf of Naples, Italy; Procedia CIRP, Volume 88, 2020, Pages 554-559

List of Abbreviations

ABC	Approximate Bayesian Computation
AI	Artificial Intelligence
AOI	Automated Optical Inspection
ANN	Artificial Neural Network
AWS	Amazon Web Service
BCD	Bayesian Change point Detection
BnB	Branch-and-Bound
CBR	Case Based Reasoning
CIM	Computer-Integrated Manufacturing
CPD	Change point detection
CPS	Cyber Physical System
CRM	Customer Relations Management
DaaS	Data-as-a-Service
DIN	Deutsches Institut für Normung
DoS	Denial of Service
EDD	Earliest Due Date
EPSRC	Engineering and Physical Sciences Research Council
ERP	Enterprise Resource Planning
FCFS	First Come First Served
FDD	Fault Detection and Diagnosis
FD-HM	Fault Detection and Health Monitoring
FoF	Factory of the Future
FFT	Fast Fourier Transform
GA	Genetic Algorithm
HV-LC	High Volume and Low Complexity
I4.0	Industry 4.0
ICT	Information and Communication Technology
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
ILP	Integer Linear Program
IoT	Internet of Things
LSB	Least Significant Bit
LP	Linear Program
LV-HC	Low Volume and High Complexity
M2M	Machine to Machine
MAS	Multi-Agent System
MCMC	Markov Chain Monte Carlo
MES	Manufacturing Execution System
MILP	Mixed-Integer Linear Programming
MV-MC	Medium Volume and Medium Complexity

Continued on next page

Continued from previous page

NUTS	No-U-Turn Sampler
OBCD	Online Bayesian Change point Detection
OPC-UA	Open Platform Communications Unified Architecture
PaaS	Platform-as-a-Service
PCA	Principal Component Analysis
PCB	Printed Circuit Board
PLM	Product Lifecycle Management
PPP	Public-Private Partnership
PRBM	Priority Rule Based Method
PROAD	Process Advisor
PyMC3	Python package for MCMC programming
QMS	Quality Management Systems
RAMI4.0	Reference Architecture Model Industrie 4.0
VM	Virtual Machine
VR	Virtual Reality
SM	Surface Mount
SMC	Sequential Monte Carlo
SME	Small and Medium Enterprises
SoA	State of the Art
THT	Through-Hole Tech
TSP	Travelling Salesman Problem
UAV	Unmanned Aerial Vehicle
WIP	Work In Progress

List of Notations

Chapter 3

$A_{clearance}$	Gate valve clearance area
r	Radius of gate valve gate and pipe
d	Distance between the gate valve gate and the pipe centre points
r_{clear}	Radius of the clear portion of the pipe

Chapter 4

x, y	Decision variables for a generic MILP formulation that take the form of column vectors $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ and $\begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix}$
c, h	Objective coefficients for a generic MILP formulation that take the form of row vectors $(c_1, \dots, c_n)$ and $(h_1, \dots, h_p)$
n, p	The maximum size of the decision variables and objective coefficients x and y respectively.
$s.t.$	Such that
A, G	Matrices used to typically denote the resources consumed or requirements satisfied by a particular decision in a MILP
b	A column vector that typically represents the requirements or amount of available resources defined by a problem and takes the form $\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$
X_{jlt}^m	A sparse, binary 4D array populated such that it contains a 1 if a machine m is processing a job j of type t at position l and 0 otherwise
Y_l^m	A matrix that contains the values C for a product completed on machine m at position l
$m \in \mu$	Machine m from a set of machines μ
$j \in J$	Job j from a set of jobs J , where a job is the process a machine performs on a product and J is equal to the total number of products being scheduled.
$l \in J$	The position l of a job j on a machine m , where the maximum number of available positions J is equal to the total number of products being schedules
$t \in T$	The type of a product t from a set of T product types
D	The due date of a product
R	The release time of a product, or the earliest time at which a product is available for processing
C	The completion time of a product

Chapter 5

Chapter 5.2

A	An event, representing an unknown or unobservable quantity or parameter of interest
B	An event, representing evidence or newly acquired data
$P(A)$	The probability of observing event A
$P(B)$	The probability of observing the event B
$P(A B)$	The conditional probability of event A taking place, given that B is true
$P(B A)$	The probability that B would be observed, given that A true
$N(\mu, \sigma)$	A stationary Guassian/Normal distribution with a mean value of μ and a variance of σ^2
$i \in J$	A point i from the full set J
$\mathbf{d}$	All observed data points
d_i	Any single data point
t	The time at which a data point d_i is observed
τ	A change point
Y	A dependant variable
$l \in m$	A coefficient number l from a possible set of m
X_l	A predictor variable
β_l	A l th regression coefficient
ϵ	A random error term
$f()$	A function describing the set of data contained between the brackets
$i : j$	A subset of data points from a full set
n	A point within a set of data
k	The number of change points
F	The data set
N	The most recently added data point

Chapter 5.4

A	The amplitude of the vibration data
V	The full range of valve states
v	A single valve state
τ	A change point

Chapter 5.5

F	The frequency of the pump vibrations
V_l	The level of valve closure at state l

Contents

Abstract	i
Acknowledgements	iii
List of Publications	iv
Abbreviations	v
List of Notations	vii
1 Introduction	1
1.1 Motivation and Scope	1
1.2 Aims and Objectives	3
1.3 Outline	4
2 Literature Review	7
2.1 Digital Transformation in Industry	7
2.2 Scheduling	15
2.3 Fault Detection and Health Monitoring	23
2.4 Research Focus and Areas of Interest	30
2.5 Knowledge Gaps	31
2.6 Summary	32
3 Research Approach	35
3.1 Introduction	35
3.2 Problem Definition	36
3.2.1 Key Challenges	37
3.3 Definition of Validation Methods	37
3.4 Research Methodology	39
3.4.1 Modular Production Line Scheduling Algorithm	40

3.4.2	Fault Detection-Health Monitoring	41
3.5	Validation and Implementation	47
3.6	Summary	48
4	Modular Production Line Scheduling Algorithm	49
4.1	Introduction	49
4.2	MILP Background	51
4.3	Problem description	55
4.4	Problem Formulation	58
4.4.1	Single machine formulation and tutorialisation	60
4.4.2	Serial machine formulation	65
4.4.3	Parallel machine formulation	68
4.5	Example data set results and discussion	69
4.6	Concluding Remarks	75
5	Bayesian Change point Detection for Fault Detection and Health Monitoring	76
5.1	Introduction	76
5.2	Bayesian Change point Detection Background	78
5.2.1	Overview of Bayesian Statistics	78
5.2.2	Bayesian Change point Detection	82
5.3	Test bench and data collection	87
5.4	Offline BCD	89
5.4.1	Formulation	89
5.4.2	Results and discussion	91
5.5	Online BCD	96
5.5.1	Formulation	96
5.5.2	Results and Discussion	99
5.6	Concluding Remarks	102
6	Industrial Application and Validation	104
6.1	Introduction	104
6.2	Modular Production Line Scheduling Algorithm Validation . . .	106

6.2.1	Methodology	106
6.2.2	Results and Discussion	107
6.3	BCD Fault Detection and Health Monitoring Validation	109
6.3.1	Background and Methodology	109
6.3.2	Results and Discussion	117
6.4	Integration with IIoT and MindSphere	125
6.4.1	Combined scheduling and FD-HM	129
6.5	Concluding Remarks	130
7	Conclusions and Future Work	132
7.1	Review of Contributions	132
7.1.1	Modular Production Line Scheduling Algorithm	132
7.1.2	Bayesian change point Detection for Fault Detection and Health Monitoring	135
7.1.3	Industrial Application	137
7.2	Future work	140
	Reference List	142

List of Tables

4.1	Sample Product Batch	61
4.2	X Array for the Sample Product Batch	62
4.3	Y Vector for the Sample Product Batch	63
4.4	Sample Product Batch C and D values	63
4.5	X Array for the Sample Product Batch after Optimisation	64
4.6	Y Vector for the Sample Product Batch after Optimisation	65
4.7	Sample Product Batch C and D values after Optimisation	65
4.8	Single Machine Results	72
4.9	Serial Machine Results	73
4.10	Parallel Machine Results	73
4.11	Production Line Simulation Results	74
6.1	Validation Simulation Results	108
6.2	Validation Simulation Results: Production line setup	108

List of Figures

2.1	Overview of change point detection algorithms	27
3.1	Research Methodology	39
3.2	Simplified Production Line Setup	40
3.3	Wave Soldering Machine Process Diagram	43
3.4	Area of overlap between two circles of different sizes	44
3.5	Test Bench Schematic	44
3.6	Test Bench Photo	45
3.7	Test Bench Data Flow	45
3.8	Pipe Blockage Rate Comparison	46
3.9	Flow Rate vs. Valve Closure	46
4.1	Transport Trolley: Batch Transport Mechanism	56
4.2	Production Line Setup	57
4.3	Visual Representation of MILP Arrays	60
5.1	FFT of pump vibrations while the valve is open	88
5.2	FFT of the pump vibrations while the valve is closed	89
5.3	Data Extraction from Recorded Pump Vibrations	90
5.4	Posterior Distribution of Amplitude Parameters	92
5.5	Posterior Distribution of Frequency Parameters	93
5.6	Expected Amplitude Values	93
5.7	Expected Frequency Values	93
5.8	Flow Rate and Predicted change points Against Valve State . .	94
5.9	Data Extraction from Recorded Pump Vibrations (Extended) .	97

5.10	Predicted change point based on Extracted Amplitudes	100
5.11	Predicted change point based on Extracted Frequencies	101
5.12	Predicted change points plotted against the Flow Rate	101
6.1	Wave Soldering Line Processes	110
6.2	Notable Faults Caused By Wave Soldering Process	111
6.3	Accelerometers on the Machine Pumps	112
6.4	Features extracted from half-hourly samples collected from Machine Pump 1	114
6.5	Artificial Data set Generated to Assess Online and Offline BCD Performance	116
6.6	Offline BCD Analysis on Filtered Machine Pump 1 Data	118
6.7	Offline BCD Analysis of Artificial Data set	120
6.8	Supplemental Offline BCD Analysis of Temporary Anomaly Data set	121
6.9	Online BCD Analysis on Filtered Machine Pump 1 Data	122
6.10	Online BCD Analysis of Artificial Data set	124
6.11	Online BCD Analysis of Pump Response During Temporary Anomaly	125
6.12	Topology Diagram: Information flow from machine to cloud	126
6.13	Topology Diagram: Supplemental data collection	127
6.14	Information Sharing Between Scheduling and FD-HM	130

Chapter 1

Introduction

1.1 Motivation and Scope

This project is an EPSRC (Engineering and Physical Sciences Research Council) iCase with Siemens titled: “Factory of the Future – Flexible, adaptable and self-optimised manufacturing systems”. Part of the project’s goal is to aid the approach to I4.0 Vision 2020+ targets for the Siemens factory. The overall objective, however, is to generally improve factory function, ideally using data already being collected or easily obtained and with a minimum of disruption to the processes currently taking place.

The factory in question manufactures a wide variety of variable speed drives, both from a catalogue of existing products and custom orders designed suit particular demands. A number of advanced digital technologies are used to streamline the design and production process, including Virtual Reality (VR) environments to aid design, and quality validation processes built into the tools used for assembly. Products are tracked from the point of order through the journey across the factory. Initial forays into the use of an Industrial Internet of Things (IIoT) network to collect data from on-site machinery have taken place, but have been limited to small-scale local applications because of the lack of native machine connectivity. Little to no analysis has been performed on the

collected data to generate insights into the production process. Production order is determined by a basic scheduling process performed once per day, which is capable of producing a “good” schedule that does not fully account for some of the complexities of a real world manufacturing environment.

The current scheduling method uses a simple Priority Rules Based Method (PRBM) to generate a production schedule based on a series of rules and priority weightings. It only takes minimal account of the sequence-dependant setup times present in Printed Circuit Board (PCB) manufacture through the use of basic type-grouping rules with a low priority. These setup times take place after a product of one type is processed by a machine, and the following product has a different type. The machine requires some amount of time to prepare (set up) to process the incoming product, which, when performed too often, can take up more than 20% of the available machine capacity [1]. However, despite its relative importance to production, it has been noted that over 90% of scheduling literature ignores setup procedures [2], [3], [4]. This presents the opportunity to create an improved scheduling method that generates a schedule through optimisation and fully accounts for sequence dependant setup times.

While some limited IIoT and cloud technologies are currently in use, a large portion of the machines in the factory have no capacity for connectivity. Upgrading or replacing this machinery would be costly and disruptive to production. However, the lack of machine connectivity also means that it becomes harder to centrally monitor and control the processes taking place, and in some cases can form data silos that can hamper data integration efforts. The machines themselves can also come with the attendant difficulties of harsh working environments and limited space for the installation sensors, requiring indirect monitoring. This necessitates the development of a low cost means of indirectly monitoring machines and thus connecting them to the IIoT network. The diversity of machine types motivates the subsequent development

of a generic means of analysing the data collected.

1.2 Aims and Objectives

The aim of this research is to find means of non-disruptively improving factory processes by leveraging new and emerging digital technologies. An analysis of an existing factory site (Siemens Congleton) has been performed. This factory is generally considered representative of the current State of the Art (SoA), especially in the field of digital manufacturing [5],[6]. Despite this, the analysis identified scope for improving the current scheduling system in use. Additionally, some existing machinery has limited capability to communicate its current health, having a limited capacity to communicate with existing infrastructure. In some cases, the only available means of fault detection is reactive and through the use of product inspection after processing. Solutions to both problems will be validated using information from and inspired by the existing factory environment. It is also intended that the developed solutions will be generic such that their application across different machines and production lines will be possible, as these issues have been shown by current research to be common across manufacturing environments.

In order to achieve this aim, the following research objectives have been identified:

1. To explore and define possibilities for measurable improvements to factory processes through digital technologies with minimum disruption to manufacturing processes.
2. To present suitable analogues for a factory environment that will facilitate initial validation of the presented contributions.
3. To design a modular scheduling algorithm that effectively accounts for sequence-dependant setup times during the manufacturing process.

4. To empower readers to formulate scheduling solutions for their own specific requirements.
5. To identify a means of exploiting available machine data at a low cost and creating the opportunity for IIoT integration on a legacy machine.
6. To provide a generic solution for Fault Detection and Health Monitoring (FD-HM) applicable to machines across a production line.

1.3 Outline

This thesis comprises of 7 chapters, which are organised as follows.

Chapter 1 - Introduction: This chapter briefly outlines the background, motivation and scope of the thesis. It presents the aims and objectives that the thesis intends to meet. It also provides a short summary of the final contributions.

Chapter 2 - Literature Review: This chapter presents and reviews the current State-of-the-Art in existing literature and industry. It covers digital transformation, Industry 4.0, relevant scheduling techniques, Fault Detection and Health Monitoring and Change Point Detection. Existing knowledge gaps will be highlighted and used to define the contributions made by this thesis.

Chapter 3 - Research Approach: This chapter presents the overarching research methodology and provides a definition of the main problems presented by the Siemens Congleton site. A simulated representation of a production line is defined to facilitate a comparison between the current scheduling method and the one presented in the thesis. The wave soldering machine as a use case for the FD-HM methods is presented. A test bench intended to simulate the behaviour of the wave soldering machine is shown alongside the kit developed to collect data from it.

Chapter 4 - Modular Production Line Scheduling Algorithm: This chapter presents the first contribution of the thesis: a flexible scheduling algorithm for solving problems including sequence dependant setup times. Three versions of the algorithm are defined, capable of generating schedules for single, serial and parallel machine setups. The single machine version is tutorialised to allow users to more easily understand how to create algorithms suitable for their specific problems. Initial assessments of each of the three algorithms are performed using sample product batches. A schedule of a full production line is also created and compared to a schedule generated by a simplified version of the existing scheduling method. The presented scheduling algorithm is shown to perform better the existing algorithm by a majority of metrics.

Chapter 5 - Bayesian change point Detection for Fault Detection and Health Monitoring: This chapter first presents the offline BCD algorithm for FD-HM. It is used to retroactively analyse data collected from a test bench analogue for the wave soldering machine use case. The offline algorithm successfully identifies the change point in both data sets used.

The online BCD algorithm is then presented and assessed on an expanded test bench data set. It provides feedback in real time about the machine state based on a comparison of current data with data previously collected during the current operating cycle. It is shown to also correctly identify the change point in both data sets and successfully infers behaviour models for the pump.

Chapter 6 - Industrial Application and Validation: This chapter presents the validation of the contributions and a discussion about the requirements for industrial application. The scheduling algorithm compares favourably against the existing factory scheduling method when anonymised production data is used. An initial analysis of the data collected from the wave soldering machine is presented, as well as a number of artificial data sets representative of potential machine faults. The FD-HM methods successfully identify the machine faults and the requirements for successful analysis of machine data are

discussed. Finally, a combined scheduling/FD-HM approach is discussed.

Chapter 7 - Conclusion and Future Work: This chapter provides a summary of the key points of the research. There is also a discussion about the future work required to develop and implement the presented contributions in a factory environment.

Chapter 2

Literature Review

2.1 Digital Transformation in Industry

Currently, a trend in many industrial and commercial environments has been digital transformation: taking and converting a variety of physical and analogue assets and making them digital, or otherwise connecting them to the internet [7]. This has had a variety of effects, one of which is improved data collection and storage capabilities through the proliferation of sensors and servers. The rise of increasingly powerful and smaller computers and processors has led to smarter and faster data processing. Combined with connectivity through the internet, a series of new data-based products and services have appeared.

Cloud computing was one of the first of these, referring to the sharing of pooled computing resources made possible through the internet [8]. The cloud allows for increased storage, processing capabilities, access to virtual machines (VMs) and a variety of applications that otherwise might not be available as a result of limited local resources. Members of the public can use it for data storage and sharing, and businesses now exist offering access to more powerful, dynamic servers on a pay-as-you-go basis. These Platform and Data-as-a-Service (PaaS and DaaS) business models free small companies from the large

investments required to have their own servers, and larger companies from purchasing servers they might under-use.

Following the growth of cloud computing has come the Internet of Things (IoT), first described with the term “ubiquitous computing” in 1991 [9]. It refers to the use of physical sensors, devices and actuators with embedded electronics that can connect to networks to exchange data [10], [11]. This includes mobile phones and wearable devices, which can gather information as they are carried round, such as their location and usage. An industrial version exists, referred to as the Industrial Internet of Things (IIoT) [12], [13], which takes advantage of the increasing affordability, intelligence, and small profile of the sensors and processors. It is intended to streamline processes and enable better product tracking, information gathering and data analysis, with an additional focus on machine to machine (M2M) communication. This differs from the consumer version mainly in its more stringent requirements for speed, security and reliability, as well as focusing more on cost reduction than revenue generation.

From an industrial perspective, the link between these two has come in the form of a number of industrial cloud platforms and the Industry 4.0 (I4.0) initiatives, such as the SmartFactoryKL [14], [15], a manufacturer independent demonstrator and research platform, or the Factory of the Future (FoF) Public-Private Partnership (PPP) [16], an initiative aimed at helping EU manufacturing enterprises adapt to global competitive pressures. Initially mentioned in Germany during the Hannover Fair in 2012 and outlined the following year as the Industrie 4.0 initiative [17], I4.0 describes the use of a wireless network of sensors and devices using an expanding collection of data to feed into manufacturing machines, allowing them to perform with greater autonomy. The concept also envisions more support for the workforce through wearable devices and collaborative robots, and an overall higher skilled workforce performing tasks such as robot coordination as machines take over a greater number of

manual jobs [17], [18], [19], [20], [21]. The increased autonomy is meant to lead to greater factory productivity and more flexible employment options for the workforce.

Industry 4.0 builds on the foundations laid by Computer-integrated Manufacturing (CIM) in the 1970s [22]. CIM was a comprehensive approach to the integration of hardware and software across the entire manufacturing process, from initial product concept to marketplace distribution, using a wide variety of technologies to automate many of the previously manual processes [22], [23]. This led to a significant improvement in manufacturing quality and quantity, capable of providing greater flexibility, reduced response times and facilitating improved levels of fault detection and health monitoring. Successful implementation was estimated to take 5-8 years [24] and required long term investment, expensive equipment and significant restructuring of a business to fully take advantage of the changes. CIM was, in some cases, considered "doomed to fail" as a result of the complexity of the required automation technology [25], and I4.0 is presented as the development of CIM that complements and facilitates it through modern Information and Communication Technologies (ICT) [26].

Industry 4.0 is currently being developed by both research groups and industry in a variety of forms in different countries, including Industrie 4.0 in Germany, America Makes in the USA and Made in China 2025 in China [27]. As a result of these disparate efforts, there is a level of risk present that solutions offered by different sources will not be compatible. For that reason it is important that new standards are established. Architectural models for I4.0 have been provided by the RAMI4.0 (Reference Architecture Model Industrie 4.0) [28] and IIRA (Industrial Internet Reference Architecture) [29], which incorporate existing standards with the goal of making them applicable for future implementation. Sticking to a series of established standards will ensure that all progress made is aligned and contributes to the growth of the technology. There is also

a trend towards having a “wrap and re-use” approach over completely removing old machines in a “rip and replace” manner [12]. Such an approach is far less expensive both in terms of capital expenditure and losses resulting from production downtime, and significantly benefits Small and Medium size Enterprises (SMEs), helping them avoid costly investments like new equipment. Retrofitting extant machines in this manner is considered an effective route into I4.0 and a means of facilitating the digital transformation of a business [12]. Research in this area is still in the early stages of architecture design [30] and the use of bespoke solutions [31], with only a limited amount of literature available that largely does not accurately reflect the digital maturity of most SMEs [32]. Existing initiatives such as Digital Manufacturing on a Shoestring [33] are working towards making these technologies more readily available to SMEs so that they can more fully benefit from I4.0 technologies.

Extracting more value from data can also take the form of data integration [34]. Data integration is a term that encapsulates the process of combining data from a number of different sources into a unified view to make it more actionable and valuable to those accessing it. This offers a number of significant benefits to users. Through the unification of data collection, access and analysis, collaboration is made easier and more effective. Time spent searching for the right data is reduced when all available data can be readily accessed, and knowledge about the available data means that rework can be avoided. Finally, the overall data quality can be increased as quality issues are identified over time and the necessary improvements can be made.

Implementing data integration within a company requires the development of four key elements [34]. These are: facilitating self-service access to data, ensuring pervasive data quality, providing a unified data collecting environment and creating a solution that can run natively on leading platforms. Self-service access to data will allow users to find and exploit existing information quickly and efficiently, avoiding time lost due to barriers such as department policies.

High data quality across the company ensures that data being used can be trusted and will require less additional processing, filtering or manipulation during analyses. A unified data collecting environment lowers the Total Cost of Ownership (TCO) of the data and allows for the automation of the data life cycle. Finally, guaranteeing that the solution can run natively on leading platforms enables implementation across the entire organisation, without the need for costly replacements to work devices. Successful data integration and subsequent exploitation can lead to benefits as great as a 93% increase in profits, 82% more sales and a 112% increase in the speed of sales growth [35]. Implementation across a wide variety of industries, including food and beverage, energy and power, and health care, demonstrates the value and flexibility of data integration.

A major stumbling block for full data integration is the data silo [36], [37], [38]: a segregated group of data stored such that it is not easily or fully accessible by other groups. Data silos can form as a result of a number of factors, the most common including company structure and legacy hardware or software. The former is especially present in larger, older companies, in which different departments would have had specific policies, procedures and goals, and designed their data storage around these. There are cases in which this is done intentionally: access to data in silos can more easily be regulated, especially in the case of companies with international bases. As such, a data silo can be used to aid in the regulation of export controlled data. Legacy hardware can create local, physical silos through a lack of connectivity while software using proprietary code and file types requires specialist tools and knowledge to take advantage of. This can lead to as much as 55% of company data becoming inaccessible for decision making [35]. Data silos limit the visibility of available data, reducing the scope of potential analyses. They can also be responsible for inefficient rework and redundant data, wasting both time and digital resources, as well as reducing data integrity in cases where the same information

is stored in multiple places but not updated at the same rate. It is worth noting here that poorly implemented IoT technologies can exacerbate the issue by increasing the number of sources and volume of data stored in segregated locations. However, current research has been developing the necessary standards required to ensure interoperability between heterogeneous devices [39] [40].

The increasing capabilities and prevalence of commercial and industrial cloud computing and IoT platforms is a key driver for data integration and digital transformation. Commercial services such as Amazon Web Service (AWS) [41] offer a variety of IoT and data management tools, as well as the ability to integrate with established and widespread apps, through a PaaS and DaaS business model. This is particularly true for Microsoft Azure [42], with the prevalence of Windows software packages and the increased use of OneDrive for data storage solutions [43]. Comparisons of these platforms against key criteria found that the choice to adopt one over the other should focus on the provider's proprietary capabilities [44]. For example, the Salesforce platform focuses on providing Customer Relations Management (CRM), making it the most suitable choice for customer engagement [45]. Users of these platforms further benefit from constant updates maintaining state-of-the-art software and expert external support that would otherwise be expensive to develop internally.

Industrial cloud and IIoT platforms have a more specific focus on data management and machine connectivity with the goal of obtaining data from and optimising processes in order to reduce costs and increase efficiency [46]. Notably, and of particular interest to this thesis, is the Siemens MindSphere platform [47], which aims to provide fast and cost effective connection of the equipment fleet alongside access to data collection and analysis. MindSphere allows for the integration of data sources across the entire business: Product Lifecycle Management (PLM), Enterprise Resource Planning (ERP), Manu-

facturing Execution Systems (MES), Quality Management Systems (QMS), Customer Relationship Management (CRM) and IoT can be combined in one analysis centre [48]. Successful implementation of this represents complete data integration, breaking down data silos and allowing for effective unification, filtering and evaluation of the data.

Alongside the benefits of digital transformation, data integration and the use of IoT and cloud platforms can present significant risks and challenges [40], the most important of which are outlined below. Chief among these are network security and privacy [17] [49], both of which are particularly vital in scenarios involving sensitive data or dangerous machinery. Failure to successfully secure networks can lead to significant damage to industrial sites [50] as well as data theft and loss of node control [51], [52], [53]. There are a number of potential attack models that have to be considered [54]. Denial of Service (DoS) attacks generally work to exhaust service provider resources and network bandwidth, but in IoT networks can further target the wireless communication infrastructure of the network and effectively “jam the channels”. Physical damage can lead to the destruction of accessible hardware. Eavesdropping attacks can passively attempt to extract data from the various communication channels in use. Finally, node capture can allow an attacker both access to information and control of a node in the network, with a scope for damage dependant on the capabilities and importance of the node. Securing a network against these attacks and ensuring privacy can be done through the integration of several mechanisms [54], [55]. The use of protocol and network security services can offer end-to-end secure communication mechanisms through the use of cryptographic algorithms and credential and key management. Identity management is necessary to for authentication and authorisation: ensuring that received data contains what it is supposed to contain, and is accessible to only the right users. Tools that allow users to manage access to their own data alongside transparency about the entities managing and using that data will

improve network privacy in general. In-built fault tolerance can help resist and prevent attacks, and allow for smoother recovery in cases where they could not be prevented. The last element to securing networks is trust in and governance of the network. Trust refers to both the trust in interactions between entities in the network and trust in the system as a whole from a user's perspective. Governance of the system offers stability, support for decision making processes and definitions of common frameworks, but has to be implemented with care to avoid an over-controlled environment.

The second major challenge to consider is network latency, particularly in time sensitive processes that require real-time feedback, rapid responses to inputs such as location tracking and awareness, or M2M communication [56]. Increasing numbers of devices on a network increase network load and detrimentally affect latency and response time. 5G standard networks are expected to maintain a latency of $<5\text{ms}$ [56], [57], which is used as a benchmark for industrial applications involving robots. Basic solutions to reducing the data load include on-site or decentralised data processing such that the transmission volume is reduced, or selective use of sensors for data collection [58]. New routing protocols with improved response times are being also trialled with the goal of reducing latency with minimal loss of information [59]. Additionally, Fog Computing is being developed as a platform providing storage, computing and network services between the cloud and end devices. This is particularly useful in situations where the full data is not required for the cloud or must be processed with very low latency and mobility [60].

Finally, the large number and heterogeneity of IoT devices and services makes interoperability a priority in this field [54]. Data and cloud integration can partly solve this problem, but in unique cases that take advantage of proprietary software and hardware, enabling inter-device communication can prove complex and costly. This can lead to the formation of data silos and prevent the full value from being extracted from generated data and analyses. Over-

coming this requires the formation and adoption of comprehensive standards to enable the creation of smart scenarios with a variety of heterogeneous devices.

2.2 Scheduling

A side effect of digital transformation and I4.0 is a trend towards “batch-size of one” production of products tailored to customers’ needs [17], [61]. As a result of this, production scheduling is becoming an increasingly complex task. This is especially true in electronics and Printed Circuit Board (PCB) manufacture: their manufacturing processes can require a lot of setup activities, which are capable of taking up more than 20% of available machine capacity [1]. Thus, reducing the amount of time spent performing setup procedures can lead to direct savings for the PCB manufacturers as well as increasing the total capacity of the factories. However, it has been noted that over 90% of scheduling literature ignores setup procedures [2], [4], [62] despite its relative importance to production. These procedures can be further broken down into setup times and costs, which have been formally defined as the time or cost required to prepare a necessary resource to perform a task [62]. Unfortunately, finding optimal solutions to these practical scheduling problems is an NP-hard problem [63], [64], [65] meaning that there is no polynomial time solution for the optimisation problem [66]. Consequently, different methods have been established to perform scheduling tasks within a reasonable time frame. The most common of these are outlined below.

Priority Rule-Based Methods (PRBMs) are a heuristic approach and are used to rapidly construct schedules for a system of machines (see [64], [67] for a list of examples and implementation). They are also referred to as dispatching rules and can be generally be classified as either local or global, and static or dynamic. Local rules consider the data of each task individually, such as in cases of single or small groups of machines. Global rules consider the data of

more than one task simultaneously or additional information not related to the task itself, useful in scheduling for a production line or entire factory. Static rules always return the same priority index, whereas dynamic rules depend on the time at which they are calculated, making the latter more valuable in scenarios which. In specific cases heuristic scheduling methods can provide optimal solutions: for example, Earliest Due Date (EDD), or Jackson's rule [68], can optimally minimise the maximum lateness in a schedule for a single machine. A large variety of possible rules exist and have been implemented both in isolation and as a combination of rules [64], [69], [70]. The effectiveness of the different priority rules (and their combinations) depends on the systems that require scheduling and the criteria used to assess the results. PRBMs are used primarily for their low complexity and ability to quickly generate a "good" schedule [63], but in most cases it is impossible to tell how close to an optimal solution the generated schedule is. They make up the bulk of scheduling solutions in traditional manufacturing [63] and are implemented through the MES layer of the factory.

Multi-Agent Systems (MASs) are a second heuristic method and have also been the subject of much research [71] to establish best practices and architectures. More recently, focus on decentralised manufacturing and I4.0 has led to a variety of MAS protocols being developed and tested [72], [73] for use in modern manufacturing environments. They are particularly valuable for determining production schedules in more dynamic environments which require adjustment to changes such as machine down time, product failures or the arrival of urgent jobs [72], [74]. The behaviours of the agents and their organisational structure determines how products travel through the system. The agents in manufacturing MASs can communicate with one another to determine the production schedule through a variety of different methods. The most common of these are negotiation protocols [73], [75], [76], which are the set of rules that govern the interactions during a negotiation. Communication can be with other agents,

such as broker/auctioneer agents in bidding based scheduling [77], [78],[79], or with supervisor/mediator agents in hybrid-based MAS architectures [72], [80]. This format is particularly suited for IIoTs as each individual agent can connect to the network and use it to communicate with the surrounding agents. However, it should be noted that the performance of a MAS is tied to the negotiation protocol in use, and it is not currently possible to analytically predict the effects of a change in the protocol [76]. Additionally, MASs have higher overheads due to the communication requirements between the agents and brokers.

Artificial Intelligence (AI) and Machine Learning (ML) is another active field in scheduling research. The first application of AI methods can be traced back to 1962 [81], where a sophisticated heuristic method capable of solving a scheduling problem with the appearance of intelligence was presented. AI methods encode knowledge about the problem domain and manipulate it through inference mechanisms to solve scheduling problems, and as such can be classified as knowledge-based systems [82]. However, they are significantly limited by an inability to adapt, leading to the inclusion of learning elements in AI systems and forming the basis for machine learning in scheduling. This takes the form of automated methods of knowledge acquisition and the integration of the capacity to improve performance based on experience. In the case of scheduling, these methods can take the form of rote and inductive learning [82]. Rote learning as a mechanism stores previously made decisions and compares them to the current scenario. The best or most suitable decision is then chosen, and the results are used to update the stored knowledge [83]. Rote learning surpasses the performance of basic heuristics but is limited to the initially chosen set of dispatching rules and system states used to train the system [82]. Conversely, inductive learning aims to infer effective rules for scheduling from training data sets [84]. Inputs take the form of production orders, the corresponding scheduling decisions made and the resultant system state,

and are analysed alongside criteria for successful inference. Inductive learning can further be refined through Case Based Reasoning (CBR) [85], which aims to exploit experience gained from past similar problem solving cases, neural network approaches [86], which can aid in the learning of appropriate criteria weights in multiple criteria decision making, and Genetic Algorithm (GA) approaches [87] to aid in the search for suitable schedules. There are two main drawbacks to the use of machine learning in scheduling: the first is its reliance on high quality, well labelled training data, which may not always be available. The amount of data required increases significantly based on the complexity of the system, and a lack of data or low quality data can significantly impact the usefulness of the approach. The second drawback is related to this in that creating a machine learning solution requires expert knowledge of and experience with the system, both to create the solution and to assess its outputs [82] [84]. It has further been noted that in many real-world environments suitable experts might not exist [88].

Optimisation approaches are more complex and time consuming, but are capable of producing a significantly better solution in more complicated scheduling cases than the methods outlined above [67]. They generally take the form of Mixed Integer Linear Programming (MILP) algorithms. A variety of models and methods for creating MILP algorithms exist in literature, addressing both specific and generalised scenarios (see for example [64], [67], [89], [90], [91], [92]). An objective (cost) function that requires minimisation or maximisation is constructed from known system characteristics, alongside a series of constraints that define the system requirements. Solutions that satisfy these constraints are most often found using Branch-and-Bound (BnB) and cutting plane techniques, which work best in combination [93] as the Branch-and-Cut method. This involves solving a relaxed version of the problem, assessing the results, and either accepting them or further refining the model until the search is complete. These can be used to find optimal solutions to the optimisation

problem. However, the complexity of the problem and the time required to solve it increases exponentially with the number of variable and constraints. Running them on a cloud computing platform with elastic computing capabilities can help avoid the prohibitively long processing times [8]. In addition to the scalability cloud platforms afford, AWS has been shown to have a number of advantages over local dedicated computing clusters, including instant access as and when required, an explicit pricing model and no additional overhead costs [94].

MILP scheduling algorithms can further be classified on the basis of the decision variables used to define them. There are 5 main types: job completion time variables, assignment and positional date variables, linear ordering variables, time indexed variables, and network variables [95]. The choice of variable defines the resultant optimisation model and each type of variable can provide different benefits such as smaller models, faster computation times or tighter lower bounds.

Job completion time is considered a key metric for the assessment of the quality of a production schedule, with a number of scheduling problem performance measures being a function of or derived from it, especially the completion time of the final job. As such, job completion time variables (also known as “natural time variables”) are considered a natural way to characterise schedules. However, job completion time variables have been known to exhibit poor performance in single machine setups and are considered too simplistic to process more complex machine environments, such as parallel setups [91].

Assignment and positional date variables specify the position of a job in a schedule and the time at which it will start processing - most often the time at which the previous job has finished [91], [95]. This type of variable is defined based on the notion that each machine has a fixed number of positions, or “slots”, into which jobs can be assigned. These positions then specify a job’s relative position to all other jobs processed on the same machine and therefore

the overall job sequence - thus creating the schedule.

Linear ordering variables, also referred to as “sequencing variables”, use binary variables to describe precedence relationships among the jobs in a set [65]. The linear ordering of a job set refers to the set of feasible solutions or permutations that the jobs can have. The binary variables are associated with pairs of jobs and specifically whether or not one precedes the other.

Time indexed variables were first introduced as a means of studying a single machine case and typically assign jobs to particular time periods [96]. These decision variables are indexed by both job and time dimensions and have since been used in a variety of machine scheduling problems. This approach leads to larger models but can provide better lower bounds than other formulations.

Network variables, or “travelling salesman variables”, take advantage of the fact that it has been shown that certain scheduling problems can be shown to resemble time-dependant Travelling Salesman Problems (TSP) [97]. Initial applications successfully modelled a single machine scheduling problem with sequence dependant setup times. Extensions taking advantage of auxiliary variables have facilitated the network based MILP formulation for parallel machine scheduling as well [98].

A comparison of models constructed using different decision variables to solve various parallel machine scheduling problems has shown that the choice of variable can significantly effect the outcome of the optimisation [91]. In all cases, computation time increases exponentially as the number of jobs and processing times increase. The model constructed with time indexed variables proved particularly effective at quickly generating feasible solutions close to optimal solutions, or yielding tighter lower bounds in situations where the allotted time constrained the optimisation. However, the computation time most suffered from large processing times, and has a difficult-to-solve LP relaxation. The model constructed with network variables was shown to be well suited for scenarios with large processing times and/or made use of release dates, but had

one of the longest computation times. The model constructed with assignment and positional date variables was capable of effectively finding feasible solutions close to the optimal or best solutions for most test cases. It suffers from a loose lower bound that slows convergence and prevents it from finishing within the allotted time but could improve through the use of additional valid inequalities. Finally, the model constructed using linear ordering variables had fairly low computation times and gave good lower bounds for scenarios with small numbers of jobs but its performance significantly degraded with increased job numbers.

The assessment and comparison of different scheduling techniques can be done through the use of a variety of metrics, dependant on the relative importance of each metric and the requirements of the production process. The most basic requirement is feasibility: the schedule must not violate constraints present in the manufacturing system in which it is to be executed [63], [99]. This refers to the constraints that define the actual processes and not those that specify levels of performance. The next is acceptability: an acceptable schedule is one that cannot be improved by trivial changes, and isn't obviously worse than any other readily available schedule [99]. The exact definition of a trivial change depends on the system in question and the complexity of the scheduling problem - for example, a trivial change might be one a knowledgeable user could make after a manual examination of the schedule.

Once a feasible and acceptable schedule has been generated, its quality can further be assessed through the aforementioned metrics. These metrics also form the basis for the cost functions used in optimisation approaches. The most common of these are described below, and they can further be used in combination and weighted to reflect their relative importance to the particular manufacturing scenario [63], [65], [64]. The total manufacturing time, or makespan, is the length of time taken to finish a set of jobs. Alongside the obvious benefits of a faster completion of a job set, a low makespan indicates

good machine utilisation. Product lateness is the difference between a product's due date and its completion time. Lateness is positive when products are late and negative when products are early: as such, a small average lateness is not necessarily meaningful. It could mean that all products were completed near their due dates, or that most products were very late and some were very early. Consequently, the average lateness and its variance are considered together, with the aim of minimising both. Similar to the lateness is product delay or tardiness, which is equal to lateness when products are late and zero otherwise. As a result, average delay is a useful measure of performance. Other metrics related to the delay and lateness include the longest delay and total number of delays, both of which a scheduling method would aim to minimise. Finally, scenario specific factors such as energy consumption can also be used as a metric for assessing a schedule.

Despite a large number of works published in literature addressing a plethora of scheduling problems, practical implementation of optimisation approaches in industry is limited and in reality many sub-optimal solutions are used instead [100]. A number of factors contribute to this lack of implementation [101], but they can generally be summarised as the inability to capture the full complexity of the entire manufacturing process. This is both due to a lack of suitable knowledge and understanding, and the difficulties present in expressing some of the aforementioned complexities through the use of a mathematical model. That said, current trends towards digital integration and transformation through the use of IoT networks and cloud platforms are a crucial step in the right direction [48], [100]. A secondary challenge is the lack of an efficient and flexible way to visualise and interact with the generated schedules, which simpler heuristic solutions often make use of [100].

Despite these difficulties, a notable example of a MILP formulation used in an industrial environment is a production algorithm implemented in a production facility for the computer technology company Dell [102], which minimised the

number of setup procedures and lead to a cost avoidance of over \$1 million annually, while successfully increasing production volume by 35%. Similarly, setup reduction techniques have been shown to be capable of providing direct benefits of \$1.8 million annually [103]. Other applications include a proposed model for a dairy plant that can reduce overall makespan by 3-5% [104], a 2% reduction in waste as well as production time in paper converting industries [105], and 20% reduction in electricity costs in a steel plant [106]. A thorough exploration of these formulations is presented in [100], where it is also noted that despite the number of cases studies, only a limited number are actually implemented in a real world environment.

2.3 Fault Detection and Health Monitoring

Fault detection and health monitoring (FD-HM) (and by extension Fault Detection and Diagnosis (FDD)) methods are also being affected by data integration and I4.0 developments. Cyber-Physical Systems (CPS) use data collected from sensors on machines to inform digital models [17] which can then be used to make decisions. This can be used to supplement existing FD-HM measures that rely upon machine and process data [107]. The “wrap and re-use” retrofitting approach [12] to intelligent machines means that additional data can be collected and improve the quality of the analysis. For example, accelerometers can be used to monitor motor performance and predict the existing faults [108]. This data can then be locally or centrally processed and presented to maintenance engineers to inform them of problems [109].

Traditional FD-HM and FDD techniques generally fall into one or a combination of the following groups: data driven, analytical or knowledge based [107]. Data-driven FD-HM techniques, such as Principal Component Analysis (PCA), depend on access to large amounts of historic data to perform statistical analyses. They can be used to accurately detect anomalies and

have the ability calculate the contributions of particular parameters to specific alerts [110]. However, in cases where the assumptions behind the analysis are incorrect, the results can show false alarms and exhibit bad detectability. Analytical approaches use detailed quantitative models to predict healthy machine behaviour and use comparison with observed data to detect faults. These approaches can prove effective when the model is accurate, but for large scale systems such a model may be unavailable or too costly and time consuming to obtain [107]. Finally, knowledge-based methods are based on qualitative models generated through causal system modelling, expert knowledge, detailed descriptions of the system or fault-symptom examples. These are well suited for fault diagnosis, but quality of the results is very dependant on the quality of the available knowledge [107], [111]. Additionally, if the machines are modified in any way it can significantly affect the reliability of the models.

More recently, research into FD-HM has been changing to take advantage of developments in I4.0. The presence of IIoT sensors and devices capable of edge computing, alongside ubiquitous cloud networks, has increased the amount of data available for processing and analysis [112]. This is reflected in a number of ways, including the increasing prevalence of AI and ML solutions better suited for processing the big data generated by the increased number of more sensitive sensors [112]. Variants of existing algorithms have been created to be more scalable and capable of both online and offline analysis [113]. However, the need for expert knowledge and industrial experience has also been highlighted, especially in the case of fault prioritisation where faults can have a high level of individuality dependant on the processes and machines being monitored [114].

The use case of a wave-soldering machine is presented in this thesis. Over time, this machine releases faulty products as a result of wave height drop due to solder build up in the pipes. There are a limited number of existing methods to measure the current wave height, such as eddy current sensors [115], laser and laser line triangulation [116], and the Solderstar Waveshuttle PRO

[117]. Eddy current and laser triangulation sensors are expensive and time-consuming to install, requiring a bespoke solution for each implementation that is handled by the end user. The Waveshuttle is a ready-made solution, but is also expensive and takes up production time to run. These solutions are specific to the machine and cannot be adapted to suit other machines in a factory environment. Additionally, their installation and use leads to interruptions in production. As a result, these methods are considered incompatible with the non-disruptive requirements set by the project brief. Other methods, such as flow rate monitoring, are also excluded because of both the harsh operating environments and the disruptive nature of the installation process of such sensors. Therefore, pump vibrations are used for the vibration analysis in this thesis because other data cannot realistically be collected. Further, vibration analysis is applicable to a large range of production machinery, making it more suitable for a generic application.

Pump vibration monitoring is not unique, and many examples can be found in both industrial and research literature [118], [119]. Techniques include comparing recorded pump vibrations against a model of the expected vibrations [120] and using pump vibrations to predict cavitation by training an Artificial Neural Network (ANN) to recognise features indicative of the fault [121]. Similarly, time-frequency analyses of vibrations have been used to diagnose wear in pump valve plates [122]. A variety of state estimation techniques that use live measurements from pumps to predict the current pump state offer trade-offs between computational time and accuracy [123]. Multiple data sources can be combined in health monitoring frameworks such as PROAD (Process Advisor) [124], which uses process, vibrational and spectral data to predict the remaining useful lifespan of a pump. Other modern approaches take advantage of IIoT devices to collect data from multiple points across a system in order to facilitate the data analysis [125]. These analyses are effective at detecting issues such as cavitation, resonance and misaligned or warped components, and

notifying operators when pump maintenance needs to take place. Approaches like these often require additional information in the form of a mathematical model of the expected pump behaviours, or labelled data sets for training the algorithms.

However, pump vibration analysis is not used to monitor general system health and predict faults that occur outside of the pump itself. This is largely assessed using pipe vibrations [126], [127], which can provide insights into general system performance, or be used to locate discrete blockages downstream of pumps [128]. The results of these approaches are affected by the locations of the sensors relative to the blockages or problem-causing elements. In the motivating scenario, access to the pipes is unavailable, and solder build-up does not necessarily form a localized blockage. Any solution needs to have low complexity so that it can be implemented on a low-cost microcomputer. Additionally, there is no preexisting vibration data with which to establish baseline readings, create a behaviour model or provide training data.

To overcome these issues, change point detection (CPD) is used, and the chosen method is assessed on a test bench. A variety of change point detection methods exists, as per Figure 2.1 [129]. CPD assesses a data set and can use a variety of methods to locate change points in the data. These are points at which a change in the state or behaviour of the data source takes place, or at which unexpected/anomalous events occur and affect the data source or the collected data. CPD algorithms will generally fall into one of two categories, from which they can be classified further: supervised and unsupervised methods. The choice of method is made based on the available data and the desired outcome of the algorithm.

Supervised CPD methods take advantage of ML algorithms to classify the different states present in a data set [129], [130]. Multi class classifiers use knowledge about all of the possible process states to locate change points at state boundaries. A sliding window moves through a data set, considering each

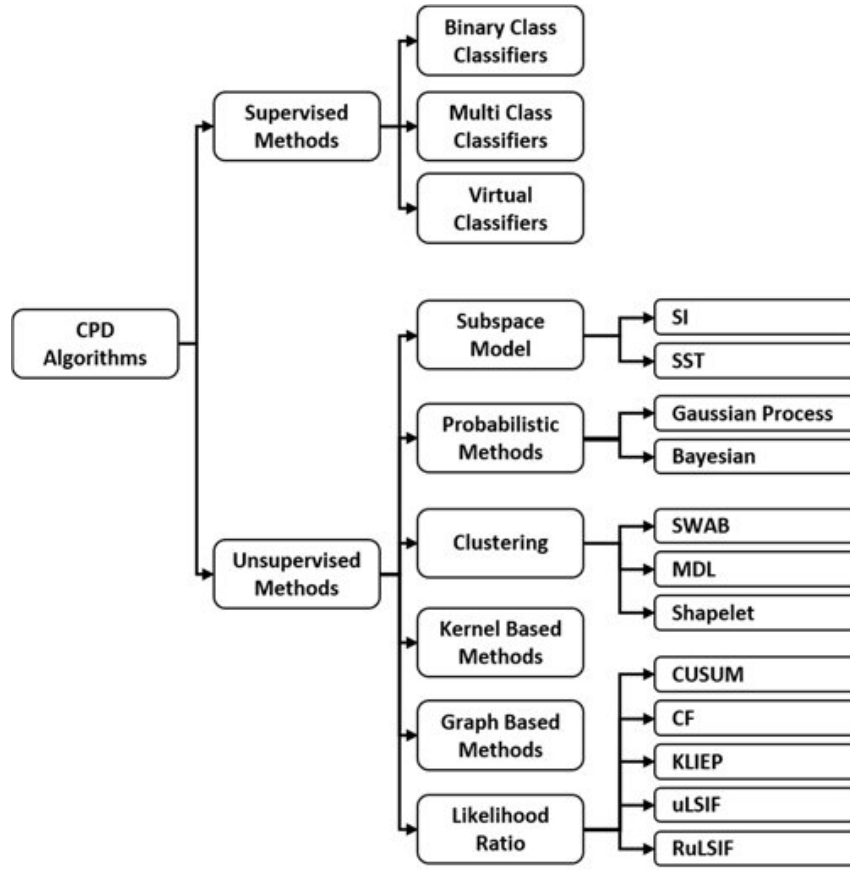


Figure 2.1: Overview of change point detection algorithms [129]

possible division between two data points as a possible boundary or change point. The training phase for this process is relatively simple but requires both sufficient and diverse data. This training data should include both each individual state class and all possible transitions between states. In some cases, if the available knowledge only covers the different states, it might still be possible to locate and classify a change point. Binary methods use a similar sliding window but make the assumption that change point or transition sequences represent one state, and sequences contained between the change points are the other. Despite the fact that fewer classes need to be learned by the ML algorithm, in cases with a large number of possible transition types the learning problem becomes more complex. Finally, virtual classifiers aim to both detect change points and interpret the change. Supervised methods tend to be more accurate if enough training data is available and the series is stationary [130].

Unsupervised CPD methods can fall into a larger range of categories and are typically used to discover patterns and pattern changes in unlabelled data, segmenting it based in its statistical features [129] [130]. Subspace modelling uses predicted state space parameters of a time series represented by state spaces to detect changes. Probabilistic methods estimate the probability distributions for each new subset of data based on the data observed since the previous candidate change point. Clustering methods form group of time series data based on their respective states and use the differences between the group features to identify the change points. Kernel-based methods map observations onto a higher-dimensional feature space and detect change points by comparing the homogeneity of each subset of data. Graph based methods use statistical tests to analyse time series observations represented as a graph. Finally, likelihood ratio methods assume that if the probability densities of two consecutive subsets of a data set are the same, they belong to the same state. Unsupervised methods are preferable in scenarios without labeled or training data or when the series is not stationary [130].

Unsupervised CPD methods are of particular interest to this thesis, as means of creating a generic approach to FD-HM for machines when there are no existing data sets and only limited knowledge about their behaviour is available. A Probabilistic method is chosen to perform the analysis, specifically Bayesian Change point Detection (BCD). BCD takes advantage of Bayesian inference, a statistical method that uses new data to inform and update assumptions made about a model. Alongside a built-in capacity to handle noisy data, Bayesian inference offers a number of additional advantages [131]:

- Bayesian model parameters and their associated functions have probability distributions, so that statements can be made about the probability of a specific event taking place.
- Bayesian inference allows the integration of prior knowledge with current results.

- Bayesian inference allows for easier predictions of observable future events based on the analyses performed.
- The results of Bayesian inference can be integrated into decision making processes with relative ease, and can provide appropriate quantifications of costs and benefits dependant on the situation.
- Bayesian inference can be performed with small sample sizes.

Once the Bayesian model of the data set has been prepared, offline BCD most often makes use of MCMC (Markov Chain Monte Carlo) methods [132], [133] to extract features from the data set that can then be used pro-actively to assess system state. However, MCMC approaches are computationally expensive and in cases that require real-time analysis, the entire data set would have to be re-analysed as new data points were collected. As such, sequential algorithms have been developed, which are capable of handling a constant stream of new data and making decisions about the occurrence of a change point based solely on the data observed to that point [134]. These algorithms form the basis of online BCD (OBCD) [135], [136].

The use case fault can be presented as an unsupervised segmentation problem as the machine transitions from a functional to a non-functional state. BCD is particularly suitable for this type of problem [137]. It provides a measurement of the probability that, based on the collected information, a change in behaviour has occurred at a certain point [138], [139]. This can then directly notify machine operators of the change in machine state, or be used to isolate the vibrational response of the pump at the point where the solder build-up is affecting the flow rate. In rare cases where the deterioration of the pipe state is instantaneous rather than gradual (such as in the case of seal failure), the change in vibrational response will still be able to inform machine operators of the need for maintenance.

Successful applications in literature include the assessment of climate records

to locate change points in climate regimes [139], detecting faults and failures in valves in an Unmanned Aerial Vehicle (UAV) fuel system [140], and detecting changes in the behaviour of a user of a text-messaging service [138]. Online applications of BCD are used to process live data [135] and predict the likelihood of newly recorded data being part of a “current run” between two change points. BCD has also been used for online signal segmentation of epileptic brain activity [141] as well as activity recognition in a home [130]. The variety of applications demonstrates the flexibility of BCD.

Alongside the number of change points, BCD can also provide an indication of the size of the changes [139], which can highlight significant events. The computational cost of this approach increases as the number of time steps increase, although this can be managed by implementing a maximum length of time or size of data set for assessment, and storing and reusing previous calculations [134]. Another drawback of this approach is a sensitivity to the prior distribution assigned to the change points present in the system, though this can be overcome by choosing unbiased distributions, using existing expertise to inform the assignments, and tuning the model based on analysis of the results.

2.4 Research Focus and Areas of Interest

The focus of this research is to find a way to improve factory processes with minimal disruption to existing processes, while also helping it approach digital transformation and I4.0 goals. The design and manufacturing methods used already take advantage of a number of modern, digital techniques. However, it was established that the current means of generating a production schedule did not account for sequence dependant setup times, which are also largely underrepresented in the literature. Additionally, limited literature is available about the application of generic FD-HM techniques in industrial environments,

particularly in situations where little to no data is available for the calibration and testing of those techniques.

2.5 Knowledge Gaps

1. Siemens' current scheduling method uses a PRBM to create a static global schedule. A number of rules are used, with the highest weighting assigned to product due dates in the form of an EDD rule. An additional priority weighting can be applied in special cases, but in most situations isn't necessary. Scheduling is done daily, in preparation for production the following day. Any failures or additional production requirements are scheduled on the following day for production the day after. This can be improved upon by using an optimisation method, which can provide a better global schedule. Constructing the global schedule from a series of modules representing local machine groups will allow for faster local rescheduling in the case of unexpected faults or machine failures, assisting the recovery process. Extant scheduling solutions for production that includes sequence-dependant setup times are limited and solve problems specific to individual cases. There is currently no available and customisable solution suitable for generating both a global and local schedule for any production line.
2. Machines currently in use in the factory have at best limited intelligence as a result of their legacy status and the expenses and difficulties involved in upgrading them. They are limited in their ability to process and respond to failures. Scheduling information is received, translated into processing data and stored locally, becoming only accessible physically. This creates a sort of local data silo and an opportunity to introduce additional sensors to supplement the information that can be gathered, as well as new ways to process that data. The application of a low cost

sensor and microcomputer can facilitate this data gathering process. The collected data can be analysed by a BCD algorithm, which will allow the features indicative of machine faults can be extracted and used to assess machine states in the future. This can inform users of the current machine state, as well as potentially the warning signs of upcoming faults. It benefits from a lack of need for preexisting data and

3. To supplement the retroactive data analysis, OBCD can be used to provide a real-time indication of the machine state. During each maintenance cycle, incoming data points are compared to the previous observed data. The likelihood of a change in machine state (and thus the occurrence of a fault) can be presented to machine operators to inform maintenance decisions and assist troubleshooting.
4. There is limited available knowledge about the retrofitting of existing machinery with low-cost sensors and processors to indirectly monitor machine health. To address this, the implementation of the presented methods in an industrial environment will be discussed, alongside means of and need for breaking down existing data silos and the security concerns facing the use of IIoT and cloud technologies.

2.6 Summary

This chapter presents a detailed literature review regarding the current trend towards digital transformation in industry. The development and use of cloud and IoT platforms has afforded industrial and commercial consumers greater levels connectivity between the machines, processes and users. Additional value can be extracted from machines, with more advanced analyses performed to better understand processes and efficiently interpret and disseminate the results to relevant users. PaaS and DaaS business models make these digital technologies more accessible to smaller companies, as does the ability to retrofit

existing machines with wireless sensors to connect them with IoT networks. This can further aid the breakdown of data silos, formed as a result of the use of legacy machines and company structure, and facilitate data integration across the company. Widespread implementation is stymied by associated risks and challenges, including security and privacy concerns which include both physical damage and data theft.

An assessment of the main methods of industrial production scheduling follows, highlighting a lack of research on sequence dependant setup times. Traditionally, scheduling in manufacturing environments is done through heuristic processes, in particular PRBMs and more recently MASs. The use of machine learning is also briefly addressed. MILP optimisation methods are discussed in greater detail as a means of fully accounting for the presence of sequence dependant setup times in manufacturing processes. The general challenges facing each type of scheduling are outlined, as well as the common means of assessing the performance of a scheduling method.

Finally, FD-HM methods were reviewed within the scope of the wave soldering machine use case. As a result of the requirements of the use case, change point detection is presented as a candidate method for detecting changes in the behaviour of the machine being monitored. The lack of pre-existing data and knowledge about machine processes, alongside the intention of creating a generic solution applicable across the fleet of machines, established BCD as the best choice for the analysis. The variety of existing applications across various fields highlighted the flexibility of the approach, although there are limited examples of industrial applications.

With this background, a number of knowledge gaps have been identified which coincide with the requirements of the project. The first of these is lack of research on scheduling with sequence dependant setup times, and the use of a PRBM scheduler in the Siemens factory. This creates the opportunity to formulate a new scheduling method, applicable across the variety of factory

machines and presented in an easy-to-interpret manner to allow for modification. The presence of high cost, high value legacy machines with no native connectivity to local IoT and cloud platforms leads naturally to the development of new data analysis and collection methods through the use of attached sensors. These have to account for the lack of preexisting data and be applicable across a variety of machines and operating conditions.

Chapter 3

Research Approach

3.1 Introduction

As part of the Siemens Vision 2020+ initiative, this research project aims to find ways of assisting the digital factory transformation with minimal disruption to the existing processes. The focus of this project is the Siemens factory in Congleton. The factory already takes advantage of many modern manufacturing practices and has a number established production lines of varying complexity. As such, problems are selected and defined with relation to the factory processes. The solutions are intended to be generic, such that they can be applied across a large number of machines. An additional goal is to ensure the contributions are non-disruptive, so that their implementation can be done at low cost without negatively affecting standard manufacturing processes or requiring expensive upgrades or replacements to machinery. The ties with the factory offer the opportunity to validate all of the contributions against the methods currently in place and/or with real factory data.

3.2 Problem Definition

The problems addressed in this thesis can be summarised as follows:

1. The Siemens Congleton factory is comprised of several main production lines, differentiated by their production volume and product complexity. The High-Volume, Low-Complexity production line (HV-LC) manufactures the largest volume of products and is made up of a set of different machine groups connected by a batch transport mechanism. Despite the high volume and variety of products, the scheduling method in place does not fully account for the sequence-dependant setup times present in during production, leading to less efficient production overall.
2. Expensive legacy machinery embedded in the production process lacks the ability to natively connect to local networks, collecting and storing information that could otherwise be used to aid in Fault Detection and Health Monitoring (FD-HM). Limited space and harsh working environments also make it difficult to attach sensors directly to sites of interest. Upgrading the machines is prohibitively costly and disruptive to the production line. A representative candidate for these issues is the wave soldering machine, which is selected as a use case for this research. This machine solders components to Printed Circuit Boards (PCBs) by passing them over solder waves kept at a specific height. The solder wave height is determined by scheduling instructions collected, translated and stored locally on the machine. Over time, solder dross builds up in the pipes, affecting the wave height and leading to the production of unsoldered products. It is currently only possible to respond to the occurrence of faults re-actively by assessing the quality of the products leaving the machine.

These problems will be presented in greater detail in the following sections.

3.2.1 Key Challenges

This section outlines the major challenges addressed by this research:

1. The lack of an effective, modern scheduling method capable of dealing with the particular machine configurations present at the factory. The current approach, a Priority Rules Based Method (PRBM), is capable of quickly constructing a "good" schedule for a set of machines or products but it is generally impossible to tell how close to an optimal solution this schedule is. It also fails to fully account for the presence of sequence dependant setup times in the production process.
2. The use case machine (a wave soldering machine) in the factory does not use the data it records in a productive manner, in the cases where it does record data. Scheduling information is converted into machine instructions and stored locally, accessible only physically. Maintenance is performed on a schedule and fault response is reactionary rather than being preemptive or predictive. The expected machine behaviour cannot be accessed in real time or directly.
3. There are limited possibilities for the assessment and integration of scheduling methods and FD-HM approaches in an existing industrial environment. A few generic benchmark tests are available in the literature but often do not reflect the reality of a production line, generally failing to account for scenarios with multiple product types, machine groups and sequence dependant setup times. There are also limited examples of real-world industrial digital integration of legacy machines.

3.3 Definition of Validation Methods

The contributions presented in this thesis are intended to be useful and applicable in industrial situations. The validation methods used are intended to

demonstrate this, and are outlined below.

Modular Production Line Scheduling Algorithm

The presented optimisation scheduling algorithm will be compared against a scheduling method representative of the one currently used by the factory. It will be assessed both in its capability to create local schedules for individual machine groups as well as a global schedule for a simulated production line, constructed from a set of machine groups. A set of different product batch sizes will be used for the individual machine groups.

The optimisation scheduling algorithm will further be assessed on its ability to generate a schedule using anonymised production data. Both local and global schedules will be created and compared to those created by a scheduling method representative of the one currently used by the factory.

Fault Detection-Health Monitoring

The FD-HM methods will be assessed based on their ability to analyse a data set generated by a test bench intended to simulate the wave soldering machine use case. The methods are offline and online Bayesian Change point Detection (BCD and OBCD) algorithms and as such they locate a change point and infer a behaviour model for the data both before and after the change point. The location of the change point and the inferred behaviour models will be compared to data set as well as the data representing the occurrence of the fault, to demonstrate whether indirect detection of the fault is possible.

The FD-HM methods will further be assessed using a series of artificial data sets representative of common machine behaviours and faults. Four data sets have been generated and the online and offline BCD analyses will attempt to locate the change points present and identify the parameters used to generate the data sets. These data sets represent: the flow rate drop experienced by the

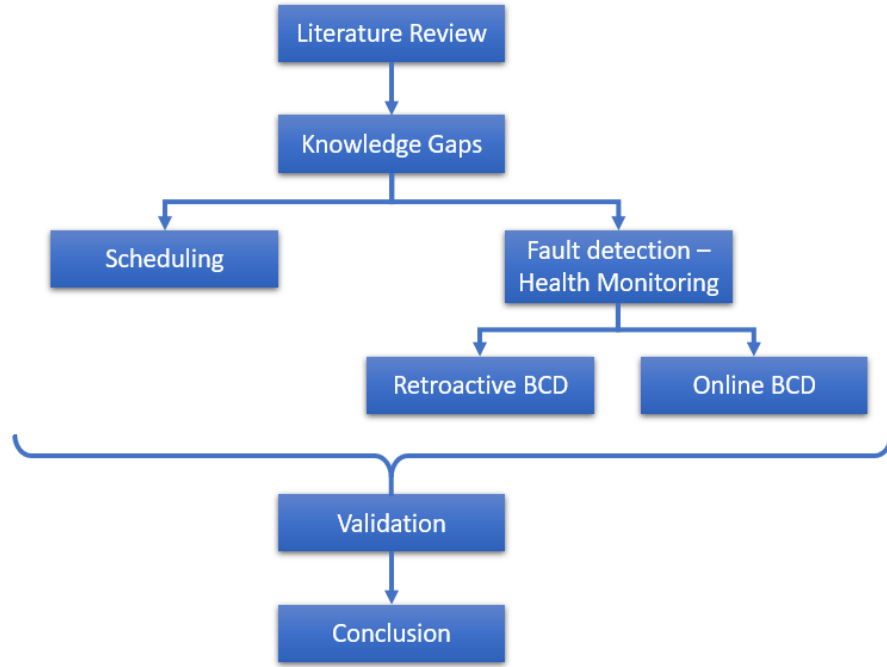


Figure 3.1: *Research Methodology*

test bench; a pump response proportional to the change in internal clearance as a result of solder dross build-up; a sudden system change, and a temporary anomalous change in output. Random noise is included in the data sets to mimic the effect of industrial noise. An initial analysis of a limited set of machine data will also be presented.

3.4 Research Methodology

The structure of this thesis is presented visually in Figure 3.1. The literature review covers the current State-of-the-Art (SoA) and identifies knowledge gaps. These are used to define the thesis contributions, which are split into Scheduling and FD-HM. The FD-HM contribution is further split into a retroactive application of BCD for fault detection, and an online application of BCD. The industrial application of the contributions is then discussed. Finally, the contributions are validated using industrial and artificial data before a conclusion to summarise the findings and future work.

3.4.1 Modular Production Line Scheduling Algorithm

The factory setup is shown in Figure 3.2, as based on a simplified version of the Siemens site. The actual setup could not be used due to data security concerns; however, this setup is considered analogous for the purposes for this thesis. This is further elaborated upon in Chapters 4 and 6. The production setup is made up of three modules representing different machine setups: single, serial and parallel machine setups, arranged in that order.

A Mixed Integer Linear Programming (MILP) approach is used because it is established to be capable of producing a better schedule than both a PRBM and Multi-Agent System (MAS) approach [142]. The MILP algorithm developed in [142] was incapable of accounting for setup times and the research itself is outside of the scope of this thesis. The extended MILP algorithm used in this thesis is presented with tutorialisation to allow industrial partners to understand the formulation process. It focuses on solving scheduling problems that have sequence-dependant setup times, of the sort encountered in this factory environment. Both a full production line schedule and individual machine setup schedules are generated, to highlight the flexibility and modularity of this approach.

MILP is slower and less responsive than PRBM and MAS approaches. Product faults and failures are dealt with on a separate production line that does not interact with the main one. Response to unplanned machine downtime is

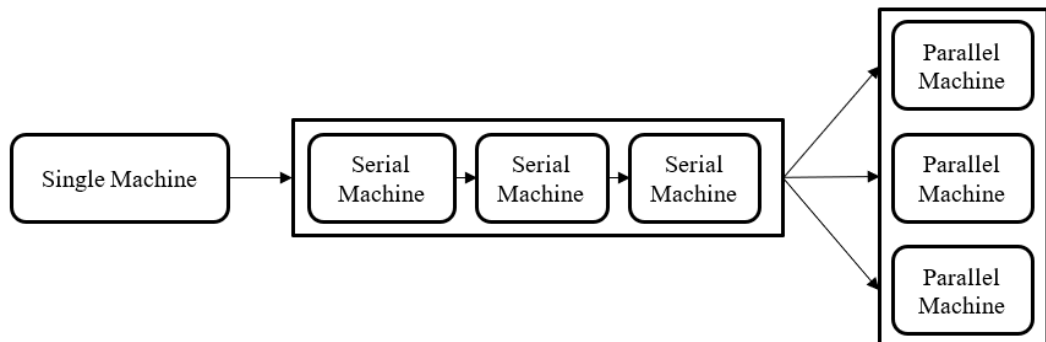


Figure 3.2: *Simplified Production Line Setup*

considered outside of the scope of this thesis. However, the slowness of the MILP approach can be overcome by leveraging the elastic computing capabilities of cloud platforms. Additionally, the approach taken here allows a global production line to be generated and in cases of disruption to the production, local schedules can be created more rapidly to facilitate rapid response and recovery.

3.4.2 Fault Detection-Health Monitoring

The use case selected for the application of the FD-HM was the wave soldering machine. This machine solders medium to large components to PCBs through the use of solder waves. Pumps move solder held in a reservoir to form the waves, which are maintained at a set height as the PCB passes over them. Over time, solder dross builds up in the pipes between the reservoir and wave nozzles. The flow rate of the solder is reduced, the wave height drops and unsoldered PCBs are produced.

Existing methods to measure wave height on a wave soldering machines limited and expensive, as well as specific to the wave-soldering process. Non-contact sensors such as eddy current [115] and laser line [116] sensors are expensive and require bespoke installation handled by the end user, as well as dedicated software to access sensor data. Contact sensors, such as the Solderstar Waveshuttle PRO [117], is another expensive piece of equipment that takes the place of a product on the production line to run. These solutions are specific to the wave soldering process and cannot be adapted to other machines in an industrial environment. In order to meet the requirements as set out in the brief, and be both generic and non-disruptive, the chosen solution has to be applicable to multiple machine types and simple and fast to install. This is further complicated by the harsh operating environment and limited space available on the wave soldering machine, a situation that is not unique amongst industrial machinery. As such, vibration analysis of the wave soldering machine pumps

has been chosen as the means of implementing the FD-HM solution.

The process diagram shown in Figure 3.3 shows how the application of the fault detection system aims to use data analytics for improved process control in the wave soldering machine. While the machine receives production instructions from the Manufacturing Execution System (MES), it translates them and stores the outputs locally. Real-time access to the data is currently impossible, and the limited space and harsh environment mean that direct sensor measurements are difficult to obtain. Disruption to the machine processes has been forbidden as part of the project brief. As such, this data analytics approach is based on non-disruptive data extraction and processing using Bayesian Change point Detection. A test bench setup has been developed to replicate the wave-soldering machine behavior as the industrial machine and environment is not suitable for development. To keep it manageable in a laboratory environment, water is used instead of solder. The difference between the vibrational characteristics of water and solder should not affect the results of the BCD so long as the deterioration of the pipe leads to a change in measured vibrations that the BCD can detect.

The test bench presented here considers the behaviour of a pump moving water through a closed system. The water tank mimics the solder reservoir, whose level is maintained through periodic additions of solder ingots. The flow rate is controlled by a gate valve, which is closed in discrete increments and goes from completely open to completely closed. It is intended to mimic the effects of solder dross build up in a wave soldering machine, which takes place over several weeks. As such, the steady state behaviour of the pump is the most relevant to this investigation. The blockage caused by the closure of the gate valve can be calculated by the following equation:

$$A_{clearance} = \pi \cdot r^2 - \left(2r^2 \cdot \cos^{-1} \left(\frac{d}{2r} \right) - 0.5d\sqrt{4r^2 - d^2} \right) \quad (3.1)$$

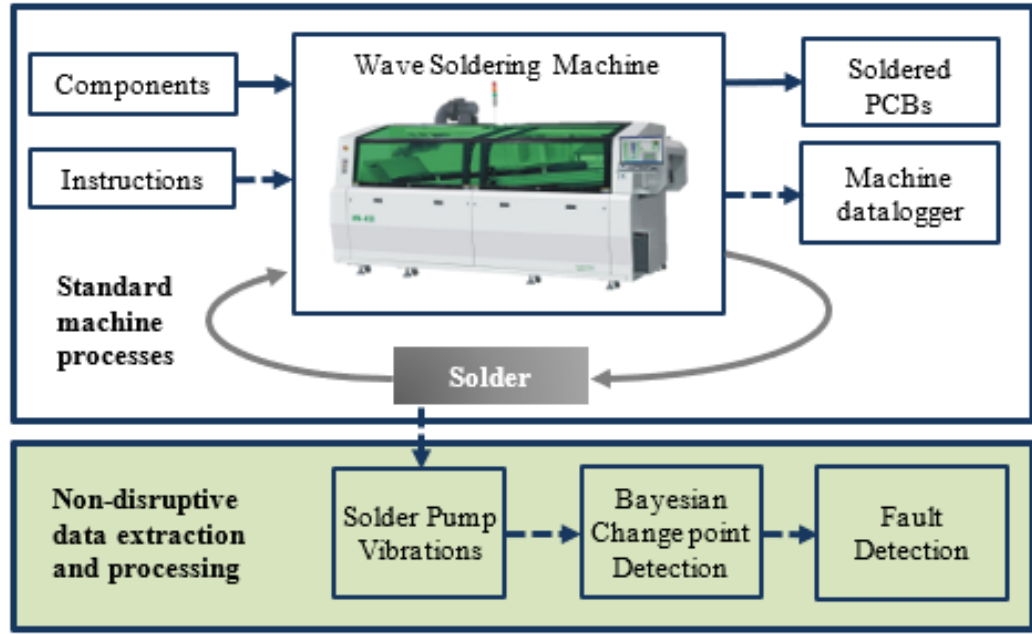


Figure 3.3: *Wave Soldering Machine Process Diagram. Dashed arrows represent the movement of data. Solid arrows represent the movement of physical parts. The green box represents the non-disruptive chapter contributions.*

which was derived from Figure 3.4 [143], simplified with the assumption that $r = R$ and $0 \leq d \leq 2r$, where r is the radius of the pipe and gate and d is the distance between the centre points of the gate valve gate and the pipe. If it is assumed that solder dross build up occurs evenly over the inside of the pipe, the blockage caused by solder dross can be calculated as follows:

$$A_{clearance} = \pi * r_{clear}^2 \quad (3.2)$$

where r_{clear} is the radius of the remaining clear pipe. Plotting these gives Figure 3.8. The test bench schematic is shown in Figure 3.5, with a photograph of the realised test bench in Figure 3.6. Figure 3.7 shows the test bench data flow. The accelerometer is mounted using bees-wax, which has a resonant frequency for 29 kHz, which is well above the frequency of the collected data and ensures that there will be no effect from it on the collected data [144].

As the valve is closed, the change in the fluid's flow rate is monitored using the flow rate sensor YF-S201, connected to an Arduino. The vibration of the

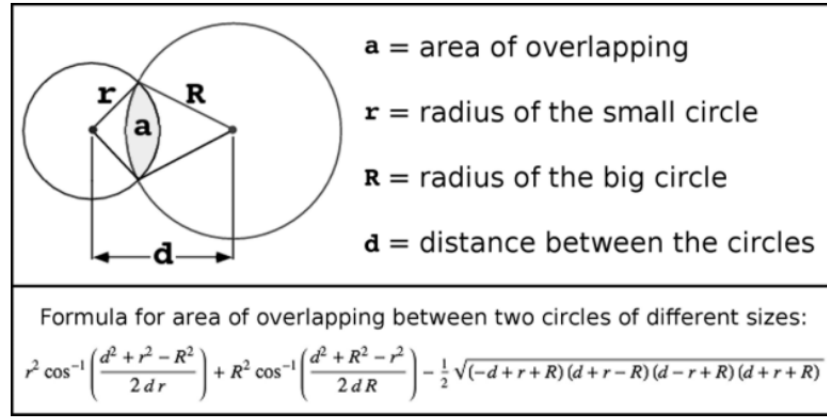


Figure 3.4: Area of overlap between two circles of different sizes [143]

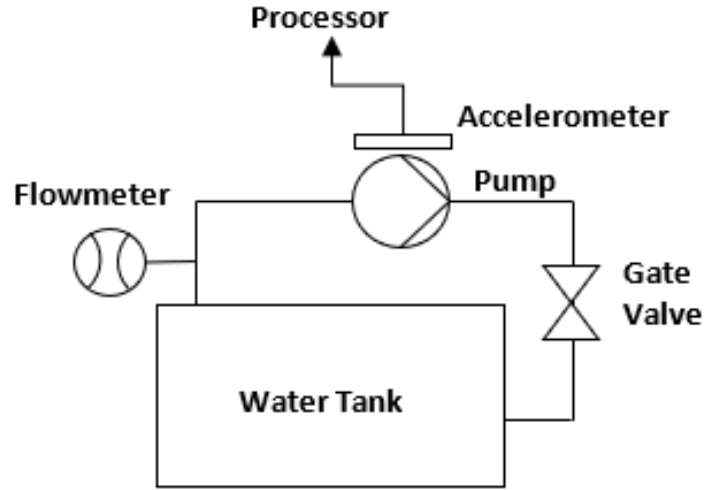


Figure 3.5: Test Bench Schematic

pump is recorded, using an accelerometer (LSM9DS1), at each increment to build up its response profile. This data is then processed at a Raspberry Pi 3B, and the results can be displayed in real time or uploaded to online networks, including any available IIoT networks. In the case of this early version of the monitoring kit, after initial data processing step, the results are stored locally for later analysis. This is a result of factory data security measures - collected data samples can not be sent off-site, so during this development period the monitoring kit will remain offline.

The change in flow rate as the valve is closed is shown in Figure 3.9. There is a small drop in the flow rate until the valve is approximately 70% before

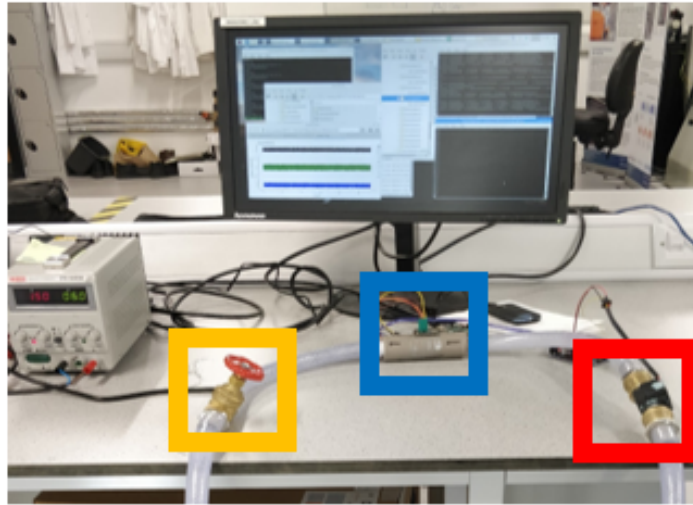


Figure 3.6: *Test Bench Photo. Yellow: gate valve; blue: pump and accelerometer; red: flow meter*

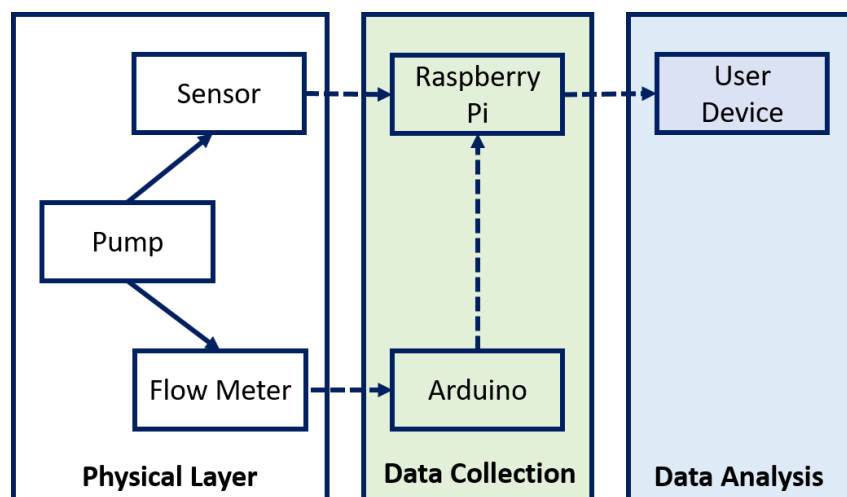


Figure 3.7: *Test Bench Data Flow. Solid lines represent physical data collection process. Dotted lines represent the movement of data*

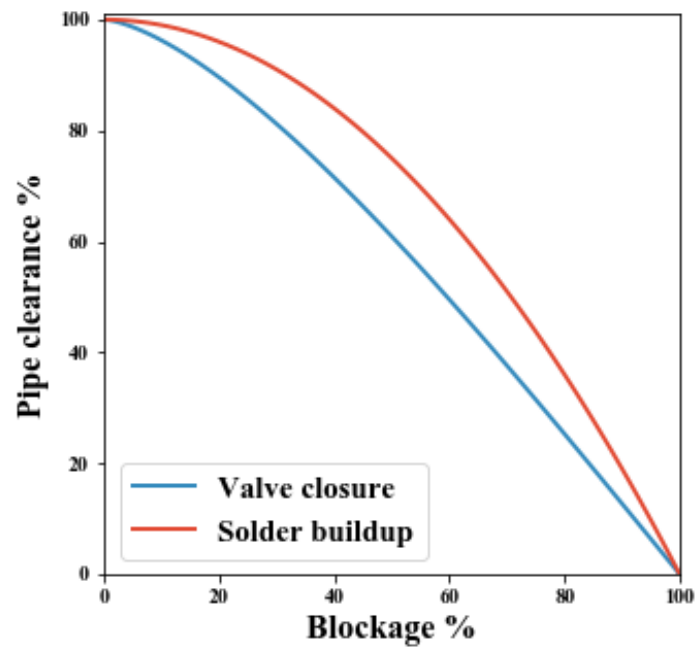


Figure 3.8: *Pipe Blockage Rate Comparison: the effect of solder build up vs. gate valve closure as a % of the pipe clearance*

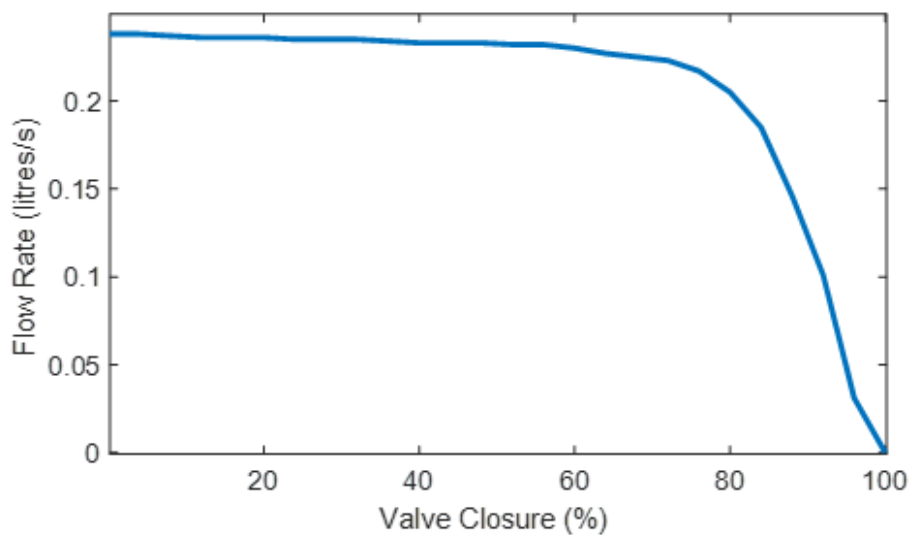


Figure 3.9: *Flow Rate vs. Valve Closure*

a transition period, and an almost linear drop to no flow from approximately 80% closed onwards. The minimal initial change is a result of the low flow rates used in this experiment. Pump vibrations are recorded with a sampling rate of 350Hz. Each collected sample is 10,000 data points in size, collected at approximately 350Hz over a period of around 30 seconds. This was determined to be the upper limit of the rate at which the Raspberry Pi could collect and store data from the accelerometer in use. The accelerometer has a range of $\pm 2g$, a sensitivity of 0.061 mg/LSB, and collects samples in the X, Y and Z axis. Each set of X, Y and Z samples is then converted into the single magnitude of acceleration value.

3.5 Validation and Implementation

In order to ensure that the methods presented here are relevant and useful, they are validated through a variety of different methods. The scheduling algorithm's performance is assessed by comparison with a representation of the existing factory scheduler, using sets of 10, 50 and 100 products released to single, serial and parallel machine groups as well as a simplified production line setup. A similar comparison will then be done using anonymised production timings.

Online and offline BCD algorithms will initially be assessed using data collected from the test bench, followed by a series of artificial data sets representative of different faults and machine behaviours. The monitoring kit will be used to collect data from the use case machine over the course of a series of maintenance cycles; however, as a result of data security concerns, the data will have to be stored locally and collected after the monitoring period. In the full application, it will connect to the local IIoT network to upload the data to the cloud platform. Finally, initial steps towards industrial implementation are discussed alongside the possibility of a combined scheduling/FD-HM approach

for production management.

3.6 Summary

By implementing a series of low-cost, minimally disruptive changes to the existing factory environment, the Siemens Congleton site can be made more efficient and competitive. The presented methods leverage modern computational techniques and state-of-the-art technology and algorithms to provide solutions to existing problems. Improved scheduling will allow the site to reduce otherwise wasted machine set-up times and decrease the overall production time of large groups of products. The application of new sensors and data processing techniques will improve the function of legacy machines, catching faults sooner and reducing the number of faulty products released in a non-disruptive manner. A combined application of the scheduling and FD-HM methods can lead to better informed schedules that work around upcoming maintenance requirements, and improved response and recovery from unexpected machine downtime.

Chapter 4

Modular Production Line Scheduling Algorithm

4.1 Introduction

This chapter presents an optimisation approach to scheduling production in a Siemens factory. Mixed Integer Linear Programming (MILP) is used to create an algorithm capable of scheduling multiple product types with sequence dependant setup times. The intention is to improve upon the existing scheduling system, which uses a Priority Rules Based Method (PRBM) to generate a daily factory schedule with minimal concern for machine setup times. Multiple machine setups are considered, as well as a combined “production line” setup mimicking factory conditions, increasing the flexibility and relevance of the approach. The construction of the algorithm is presented as a series of variants representing the different types of machine setup and tutorialised to make it more accessible . This will provide readers with the knowledge and understanding required to construct more specialised algorithms to suit specific industrial environments. The individual algorithm variants will be functional means of generating local schedules for single, serial and parallel machine setups in small-scale production scenarios or as part of a recovery and response

process in cases of disruption to production. They can also be arranged to form different production lines, an example of which will be used to assess this approach's performance. The schedules generated will be compared to a simplified PRBM representative of a standard industrial scheduler.

4.2 MILP Background

Mixed Integer Linear Programming is a version of Integer Linear Programming (ILP), which in turn is an extension of Linear Programming (LP). LP is an optimisation method used to solve problems by maximising or minimising a linear function subject to linear constraints on the variables. It is a useful tool for modelling wide varieties of problems, particularly those in planning, routing and scheduling. It is often the case in these problems that results have to be constrained to discrete values, such as a number of workers, tools, or vehicles. ILP adds the necessary integrality constraints to the variables to represent this quality [145] [146]. These integer variables can also represent decisions with yes/no outcomes: these are represented by variables constrained to values of 0 and 1, referred to as *binary variables*. In cases where both integer and non-integer variables are used, the problem becomes a MILP.

A MILP problem will generally take the following form [146]:

$$\begin{aligned}
 \max \quad & cx + hy \\
 \text{s.t.} \quad & Ax + Gy(\leq, =, \geq)b \\
 & x \geq 0 \text{ integer} \\
 & y \geq 0,
 \end{aligned}$$

where x and y are the column vectors $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ and $\begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix}$ respectively and contain the variables being optimised, referred to as the *decision variables*. In this case, the *decision variables* in x are constrained to non-negative integers, while those in y are allowed to be any non-negative real value. c and h are row vectors $(c_1, \dots, c_n)$ and $(h_1, \dots, h_p)$, the *objective coefficients* associated with either the costs or revenues being minimised or maximised by the decisions made by the corresponding *decision variables*. Column vector $b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$

generally represents the requirements or amount of available resources defined by the problem. Matrices A and G have sizes $m \times n$ and $m \times p$ respectively and typically denote how much of a resource is consumed or requirement satisfied by a particular decision.

At this point it is valuable to note that nonlinear objectives and constraints are not used in these models, prohibiting the multiplication of decision variables, the maximum of several variables or absolute values of variables. If these conditions are desired, they must be achieved with different techniques. The reason for this is that nonlinear constraints significantly increase the difficulty of the problem. It has been shown that the use of quadratic constraints in integer programming creates problems that are *undecidable* [147], which is to say that an algorithm to solve it cannot exist. Further to that, even if no solution is found after a large amount of time, it may still not be possible to conclude that none exists [146].

Defining the constraints for the ILP problem creates a set of feasible solutions: a set of values for the decision variables that satisfy these constraints. Enumeration can be used to find the solution at this point, but the size of the problem grows exponentially with the number of elements and makes this impractical for problems of any significant size [145] [146]. Metaheuristic techniques (such as local search, tabu search, and genetic algorithms) can fairly quickly locate local optima, but it is often impossible to tell how close to the global optimal solution the results are. Also, in cases where no solution is found given a long time, it cannot be known whether that is because there is no feasible solution. For this reason, several exact solving algorithms have been developed. The most commonly used algorithms are the Branch-and-Bound and cutting plane methods. Furthermore, a combination of these two methods, the Branch-and-Cut method, is frequently used. All three methods are briefly outlined below.

A first step to solving a ILP problem with the aforementioned techniques

is *relaxation*: the removal of some of the restrictions or constraints. The most common of these is the *LP relaxation*, the elimination of the integrality constraint. This turns the NP-hard ILP problem (which cannot be solved in polynomial time [66]) into a LP problem that can be solved in polynomial time [93], the solution of which can then be used to gain information about the solution to the original problem. LP solvers generally fall into two types: simplex algorithms [148] and interior-point methods [149], which are currently considered similarly efficient for routine applications [150]. This presents a naive solution to the ILP: rounding the solved LP relaxation to the nearest integer values. However, the results of this relaxation can often be a long way from the actual solution, or in some cases break other constraints [145].

Branch-and-bound [151] uses the solution of the linear relaxation to determine the upper bound of the objective value and the associated decision variables. In the ideal situation, these will be integral and as such will be the optimal feasible solution to the ILP. Otherwise, the problem is split into two or more LP sub-problems with additional constraints. These new constraints usually split the feasible region about one of the original solution's fractional decision variables such that it is no longer a valid solution to the linear relaxation, but the nearest integers either side are: these new problems become the new branches. These LP relaxations are solved and the results can be interpreted in the following ways:

- If the solution is integral, the objective value becomes the new best lower bound and the associated decision variables become the incumbent solution. The branch becomes *pruned by integrality*.
- If the problem becomes infeasible, the branch is *pruned by infeasibility*.
- If the solution has a lower objective value than the current best lower bound, the branch is *pruned by bound*.
- If the solution has a higher objective value than the current best lower

bound, but is not integral, a new set of branches is created and the search continues.

This process is repeated until all branches are pruned. At this point, the incumbent solution is accepted as the optimal solution. When using branch-and-bound to solve ILPs, there are several key factors that can affect the efficiency of the algorithm [146] [151]. These include the ILP formulation itself: formulating it in a way that reduces the gap between the optimal objective value of the LP relaxation and the optimal objective value of the ILP can significantly reduce the required number of branches. Branching and node selection strategies can depend on the solver in use or can be chosen specifically for the problem. Finally, heuristics are used to quickly provide lower bounds for the objective values, and better heuristics can provide tighter bounds to prune worse branches at earlier stages. It should be noted here that while it has been shown that heuristics do not dramatically impact the ability of a MILP solver to prove optimality [152], there is a significant psychological benefit to the user to see a high quality feasible output produced early [153]. Additionally, should an algorithm be terminated before completion, the heuristic solution will ensure that the output's objective value is at least as good as that of the heuristic.

Cutting planes [154] [155] [156] take the linear relaxation and try to strengthen it by removing non-integral solutions. This is done by narrowing the search region, adding "cuts" in the form of valid inequalities that do not cut off integer solutions but do remove some fractional solutions. Finding the cutting plane is a *separation problem* that has to be solved. The aim of these cutting planes is to remove the optimal LP relaxation solution from the feasible region. This can lead to a solution to the ILP in one of two ways. The first is by including enough valid inequalities before solving the LP relaxation, such that the LP relaxation provides an integer solution. The second is by looping between solving the LP relaxation and adding new cutting planes until the LP

relaxation yields an integer solution. However, this approach can suffer from intractability or numerical instability. Producing a quality cutting plane can lead to a trade-off with the required running time [146], and it can take an exponential number of steps to approach an integral optimal solution [156].

The most effective methods for solving ILPs use a combination of both branch-and-bound and cutting planes in the form of branch-and-cut algorithms [157], [158], [159]. The linear relaxation is solved and then pruning occurs. At this point cutting planes can be added if it seems likely to strengthen the formulation. With these cuts, the relaxation is solved again and the choice to cut or not cut is made again. If not, branching is done and the process repeats. Decisions about adding new cuts are made based on the success of previous cuts, with more cuts typically being added during earlier stages of the process and fewer deeper in the tree. In cases where cuts are only used to tighten the initial LP relaxation, the algorithm is referred to as cut-and-branch.

4.3 Problem description

The motivation for this chapter is a Siemens Printed Circuit Board (PCB) factory that processes a large variety of products across a number of product families. There are three main production lines: High Volume, Low Complexity line (HV-LC); Medium Volume and Medium Complexity line (MV-MC); and Low Volume and High Complexity (LV-HC). Each production line manufactures a different set of products. Several of the processing stages include machines that are shared between the production lines. The focus of this chapter is the HV-LC line, which has a larger proportion of automated and predictable processes.

The HV-LC production line is composed of a number of discrete stages, connected by several different transport mechanisms. The first stage involves the placement of small components on a blank PCB by a Surface Mount (SM)



Figure 4.1: *Transport Trolley: Batch Transport Mechanism*

machine, a single machine that sequentially processes individual PCBs. Upon completion, these are stacked in a trolley (Figure 4.1) and transported as a batch to the next process once the pallet is full. The machines that follow perform a number of serial processes on each product. THT (Through Hole Tech), wave soldering or selective soldering is performed to secure larger components to the PCB, followed by resin coating as the products are transported by conveyor belt. The products are then transported by trolley again to the next stage. The final mechanical assembly processes and product function tests are completed on a set of parallel machines. Several component tests are integrated into the manufacturing processes. Once the products pass these inspections, they are packaged and prepared for dispatch in another part of the factory, separate from the production line. The sequence-dependant setup processes take place after a machine processes a product of one type and the following product in the queue is of a different type.

A simplified version of the HV-LC production line described above is presented in Figure 4.2. This is the factory setup that will be used to test the scheduling algorithm presented in this chapter against a simplified version of the existing scheduling method. The first single machine represents the SM machine.

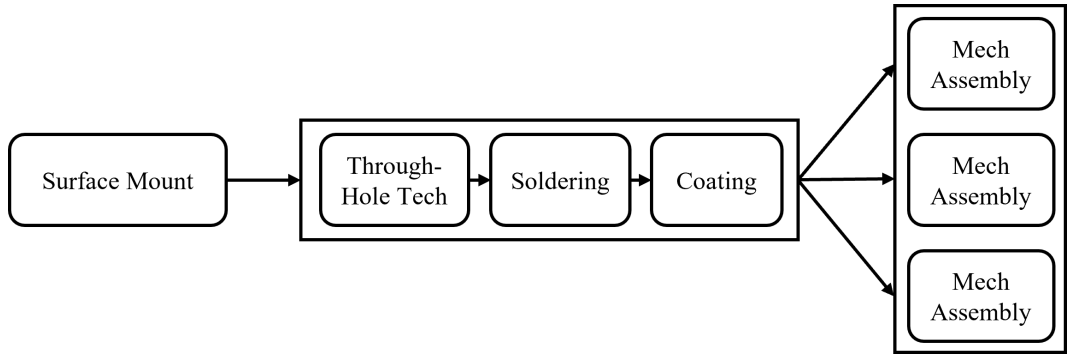


Figure 4.2: *Production Line Setup*

The serial machines represent the THT, soldering and coating processes. Finally, the parallel machines represent the mechanical assembly stations and integrated tests.

The factory's production schedules are currently prepared on a daily basis, for the following day. A combination of product quotas and custom orders determines the number and type of processes required. This schedule cannot be altered - faulty products are taken to a separate production area for repair and touch-up, while other disruptions to the schedule are dealt with by updating the schedule for the following day. These schedules are generated using a weighted PRBM which sorts products based on attributes including due dates, batch sizes and assigned priority. Rule weightings are decided upon using trial and error. This can lead to wasted time performing unnecessary setup procedures as production lines swap between processing different product types.

To assess the performance of the MILP formulation, it is compared to a simplified representation of the incumbent Siemens scheduler. This takes the form of a PRBM that uses Earliest Due Date (EDD) sorting, which, as discussed in Chapter 2, can provide optimum solutions for reducing the maximum lateness of delay in a single machine setup. For the parallel machine setup, a First Come First Served (FCFS) rule is also used, with products being sent to the machine available soonest. If there is a scenario where two or more machines are available simultaneously, the one currently set up to process the correct product type, or with the shortest current processing time, is selected.

A number of assumptions are made about the system to simplify the simulation. The following, a selection of those listed in [64] and [63], have been included as they are considered valid in most cases and relevant here:

- A machine only executes one operation at a time.
- Once an operation has been started, it cannot be interrupted - there is no preemption.
- Machines are the only limited capacity resource considered.
- Jobs are independent of one another.
- There is no cancellation of jobs.

Additionally, the time taken to transport products is considered outside of the scope of the requirements for this algorithm and is assumed to be negligible. However, the effect of the batch transport between different machines is accounted for by making the release dates of a batch of completed products equal to the completion time of the last product in the batch. Furthermore, discussion with Siemens has allowed the assumption that all machines can be considered identical for the purposes of testing the MILP formulation, with a setup time equal to half of the processing time. Likewise, unexpected machine downtime and product failure are not considered. The former occurs rarely enough that standard scheduling does not have to account for it, and the latter is dealt with outside of the standard production line, on a specialised correction and repair line.

4.4 Problem Formulation

The MILP formulation used in this chapter uses the assignment and positional date variables X and Y , represented with Figure 4.3a and Figure 4.3b. This

formulation is developed from the algorithm presented in [142], which was incapable of accounting for setup times. The original algorithm and research is not presented here as it is outside of the scope of the thesis, as it focused exclusively on comparison between different scheduling methods. The presented MILP formulation is capable of scheduling sets of single, serial and parallel machines with minor adjustments. The single machine setup is tutorialised to improve understanding and encourage implementation. It provides a schedule for multiple product types that require sequence dependant setup times. Using a cost function, the formulation minimises the sum of the maximum delay and average completion time. This can be modified to better suit the requirements of the user. The formulation's performance on each of these setups is presented alongside a combination of the three as a representation of a factory production line. This provides a realistic idea of its performance and possible shortcomings.

Assignment and positional date decision variables have been chosen as the basis for this formulation for a number of reasons. They work by specifying the position of a job in a schedule and the time at which it will start processing - most often the time at which the previous job has finished. The use of these variables creates a compact formulation that can effectively reach near-optimal results even when constrained by the allotted computation time [91]. Furthermore, it is particularly suitable for solving more complex scheduling problems that include release dates, deadlines, and precedence constraints such as those required for sequence dependant setup times [95], and can relatively easily be extended to account for a variety of machine situations. These decision variables also allow for the construction of more readily interpretable formulations. Assignment and positional date variables do generally suffer from loose lower bounds, but this will partially be addressed through the use of heuristics [91]. The variable X (Figure 4.3a) is a sparse, binary 4D array. The algorithm works by populating it so that a machine (m) is processing a job (j) of type

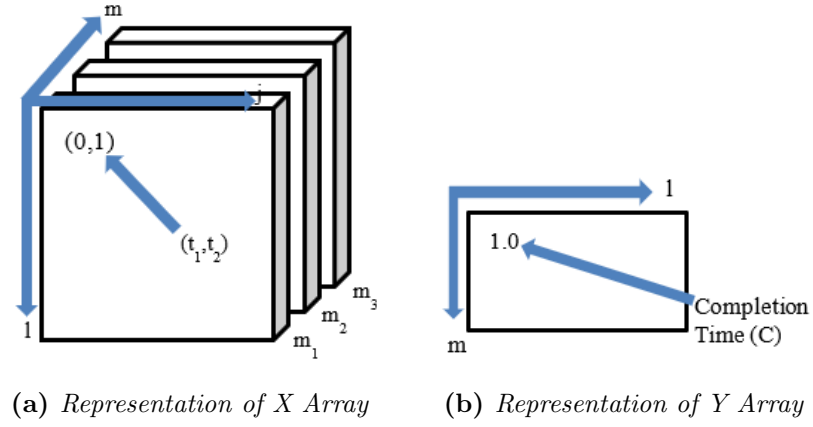


Figure 4.3: Visual Representation of MILP Arrays

(t) at position (l) if there is a 1 at that position, and a 0 otherwise. There are μ total machines and T product types. A machine can process up to J total jobs, where J is the number of products being scheduled, and each job is linked to a particular product. The position l refers to the production order on that particular machine. Each product is generated with a type (t) , and due date (D) and a release date (R) .

Based on X and the specifications of the processes and products, the completion time (C) of each product is calculated. The value of C for a product processed at position l on machine m is stored in matrix Y (Figure 4.3b). This is used by a cost function to assess the effectiveness of a particular schedule generated by the MILP.

4.4.1 Single machine formulation and tutorialisation

In this section of the chapter, a simplified version of the scheduling problem will be presented alongside the formulation to demonstrate how the MILP works. A sample batch of four products (whose details are presented in Table 4.1, in EDD order) will be used. As the formulation is explained, the effects of each of the constraints will be outlined.

The single machine formulation is the simplest version of the formulation, where $\mu = 1$, significantly reducing the size of the matrices formed. This

makes X 3D and Y a vector. The constraints in use are shown below. Let the binary variable $X_{jlt} = 1$ if job j has type t and is assigned to position l on the machine; otherwise $X_{jlt} = 0$. Additionally, variable Y_l is a non-negative positional date variable denoting the completion time of the job at position l on the machine.

$$\sum_{l \in J} \sum_{t \in T} X_{jlt} = 1, \quad j \in J \quad (4.1)$$

$$\sum_{j \in J} \sum_{t \in T} X_{jlt} \leq 1, \quad l \in J \quad (4.2)$$

Constraints (4.1) and (4.2) ensure that all jobs are assigned to exactly one position on the machine, and that each position on the machine contains at most one job respectively, regardless of job type. Based on these constraints, assuming that the MILP has been initialised with values from an EDD sorting of the products, Table 4.2 shows the resultant X array. Each row represents a particular product (row 1 represents product 1) and the columns represent the production order (column 1 represents the first product in the processing order). As the table shows, X is sparse. The table has also been simplified – any cells with no products in do not show the zeros for both product types.

$$Y_l \geq 0, \quad l \in J \quad (4.3)$$

Constraint (4.3) is the non-negativity constraint for vector Y .

Product ID (j)	Due Date (D)	Type (t)	Release Date (R)
1	1	1	0
2	2	2	0
3	3	1	0
4	4	2	0

Table 4.1: Sample Product Batch

ID	Product Order			
	$\mathbf{T1}^{\mathbf{T2}}$	$\mathbf{T1}^{\mathbf{T2}}$	$\mathbf{T1}^{\mathbf{T2}}$	$\mathbf{T1}^{\mathbf{T2}}$
1	$\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}$	0	0	0
2	0	$\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}$	0	0
3	0	0	$\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}$	0
4	0	0	0	$\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}$

Table 4.2: X Array for the Sample Product Batch

$$Y_l \geq \sum_{j \in J} (p_j + R_j) X_{jlt}, \quad l \in J \quad (4.4)$$

Constraint (4.4) ensures that jobs are not completed earlier than their processing time (p) added to the release date (R) of the associated product. It calculates the completion time of the first job (Y_1) and the initial setup of the production line with regards to that product's type. In the example presented here, $R = 0$, and $p = 1$.

$$Y_l \geq Y_{l-1} + \sum_{j \in J} p_j X_{jlt} + s_t (X_{jlt} - X_{j(l-1)t}), \quad l \in J : l \geq 2, t \in T \quad (4.5)$$

$$Y_l \geq Y_{l-1} + \sum_{j \in J} p_j X_{jlt} - s_t (X_{jlt} - X_{j(l-1)t}), \quad l \in J : l \geq 2, t \in T \quad (4.6)$$

Constraints (4.5) and (4.6) calculate the completion time of the product at position l (where $l \geq 2$) on the machine, based on the completion time of the previous product at $l - 1$ and any setup times (s). By including both positive and negative versions of the setup constraint, the order in which the products appear doesn't matter. If there is no change in type, the value inside the brackets is 0 and thus the setup time is 0. Otherwise, it will be 1 or -1. Whichever of these leads to the largest Y_l value will take precedence. Table 4.3 shows the results of these equations and what the resultant Y vector would look like using the sample product batch, where $s = 0.5$. In this case, each

column represents the production order.

Constraint	Production Order			
	1	2	3	4
4.4	1	1	1	1
4.5	-	1.5	4	4.5
4.6	-	2.5	3	5.5
Y_l	1	2.5	4	5.5

Table 4.3: Y Vector for the Sample Product Batch

$$C_j \geq Y_l - M(1 - \sum_{t \in T} X_{jlt}), \quad j \in J, l \in J \quad (4.7)$$

Constraint (4.7) provides the completion time C of the job j , where M is a sufficiently large number (with a value of 10 in this case). In this particular case, C becomes equal to Y , but in cases with parallel machines, constraint 4.7 guarantees that only the machines that have actually completed jobs by processing products are the ones recording a completion time.

$$D_j = \max(C_j - d_j, 0), \quad j \in J \quad (4.8)$$

Constraint (4.8) calculates the product delay (D) using the completion time of the associated job (C_j) the due date (d) and is used in the cost function. Products that are completed before their due date are not delayed and as such have a delay value of 0. The results of these constraints is shown in Table 4.4.

Product ID	Completion Time (C)	Delay (D)
1	1	0
2	2.5	0.5
3	4	1
4	5.5	1.5

Table 4.4: Sample Product Batch C and D values

The cost function being minimised is as follows:

$$\min_{j \in J} (\max(D_j) + \frac{1}{n_j} \sum_{j \in J} C_j), \quad (4.9)$$

minimising the sum of the maximum delay and average completion time. The addition of the average completion time prevents the algorithm from settling on a low maximum delay solution with a large average completion time. The cost function can be modified to optimise for a number of different objectives, with their own weightings depending on the requirements of the process.

In the case of the sample batch of products, when sorting using EDD, the cost function is equal to 4.75. The effect of the MILP formulation on the production schedule can be seen in Tables 4.5 to 4.7, and the result of the cost function is 4.25, a reduction of approximately 11%. From Table 4.5, it is clear that the processing order has been changed to 1,3,2,4. This leads to only one set of setup times being required, which means the overall processing time is reduced to 4.5. In addition to this, only two products end up being delayed, and the average delay drops to 0.5 from 0.75. The longest delay remains the same, at 1.5. While this particular example can trivially be solved by hand, in more complex cases solvers will generally use the branch-and-cut technique described earlier in combination with heuristic and simplex methods.

ID	Product Order			
	T1 ^{T2}	T1 ^{T2}	T1 ^{T2}	T1 ^{T2}
1	0 ¹	0	0	0
2	0	0	0 ¹	0
3	0	0 ¹	0	0
4	0	0	0	0 ¹

Table 4.5: X Array for the Sample Product Batch after Optimisation

For the sake of comparison, and to demonstrate its effects, a different cost function will be presented alongside the results of the subsequent optimisation. The following cost function will minimise the number of delayed products:

Constraint	Production Order			
	1	3	2	4
4.4	1	1	1	1
4.5	-	2	2.5	4.5
4.6	-	2	3.5	4.5
Y_l	1	2	3.5	4.5

Table 4.6: Y Vector for the Sample Product Batch after Optimisation

Product ID	Completion Time (C)	Delay (D)
1	1	0
2	3.5	1.5
3	2	0
4	4.5	0.5

Table 4.7: Sample Product Batch C and D values after Optimisation

$$\min_{j \in J} (\sum_{j \in J} [D_j > 0]), \quad (4.10)$$

Where the statement contained in square brackets, $D_j > 0$, represents 1 if it is true and 0 otherwise. The schedule generated by EDD sorting has 3 delayed products and as such a cost function of 3. The MILP formulation generates a schedule with a production order of 1,3,4,2, which has a single delayed product and as such a cost function of 1. This is an improvement of 66%. However, the total delay is now 2.5, which is a 66% increase over both the EDD schedule and the previous cost function.

4.4.2 Serial machine formulation

In this version of the formulation there are $m = \mu$ machines in series. The constraints used are similar those in the first formulation, but now include the machine variable. As such, the binary variable $X_{jlt}^m = 1$ if job j has type t and is assigned to position l on machine m ; otherwise $X_{jlt}^m = 0$. Y_l^m is now a non-negative positional date variable denoting the completion time of the job at position l on machine m .

$$\sum_{m \in \mu} \sum_{l \in J} \sum_{t \in T} X_{jlt}^m = 1, \quad j \in J \quad (4.11)$$

$$\sum_{j \in J} \sum_{t \in T} X_{jlt}^m \leq 1, \quad l \in J, m \in \mu \quad (4.12)$$

Constraints (4.11) and (4.12) ensure that all jobs are assigned to exactly one position on only one machine, and that each position on every machine contains at most one job respectively, regardless of job type.

$$l_j^m \geq l_j^{m-1}, \quad j \in J, m \in \mu : m \geq 2 \quad (4.13)$$

Constraint (4.13) is a precedence constraint. This guarantees that a job cannot take place on a machine before the prerequisite jobs are completed on the previous machines. In effect, a product cannot be processed on a machine before it has been processed on the previous machines.

$$Y_l^m \geq 0, \quad m \in \mu, l \in J \quad (4.14)$$

Constraint (4.14) is the non-negativity constraint.

$$Y_l^1 \geq \sum_{j \in J} (p_j^1 + R_j) X_{jlt}^1, \quad m \in 1, l \in J \quad (4.15)$$

Constraint (4.15) calculates the completion time of jobs on the first machine, with the inclusion of product release dates. The following machines do not need to consider the product release dates, since the products have to be processed on the first machine before moving on.

$$Y_1^m \geq Y_1^{m-1} + \sum_{j \in J} p_j^m X_{j1t}^m + s_t^m X_{j1t}^m, \quad m \in \mu : m \geq 2, l = 1 \quad (4.16)$$

Constraint (4.16) calculates the completion time of the first job on all machines following the first one.

$$Y_l^1 \geq Y_{l-1}^1 + \sum_{j \in J} p_j^1 X_{jlt}^1 + s_t^1 (X_{jlt}^1 - X_{j(l-1)t}^1), \quad m = 1, l \in J : l \geq 2, t \in T \quad (4.17)$$

$$Y_l^1 \geq Y_{l-1}^1 + \sum_{j \in J} p_j^1 X_{jlt}^1 - s_t^1 (X_{jlt}^1 - X_{j(l-1)t}^1), \quad m = 1, l \in J : l \geq 2, t \in T \quad (4.18)$$

Constraints (4.17) and (4.18) calculate the completion time of all jobs on the first machine, based on the completion time of the previous job on the first machine and any setup times. As with the single machine formulation, the inclusion of both positive and negative versions of the constraint means that the order in which the products appear doesn't matter.

$$Y_l^m \geq Y_l^{m-1} + \sum_{j \in J} p_j^m X_{jlt}^m + s_t^m (X_{jlt}^m - X_{jlt}^{m-1}), \quad m \in \mu : m \geq 2, l \in J : l \geq 2, t \in T \quad (4.19)$$

$$Y_l^m \geq Y_l^{m-1} + \sum_{j \in J} p_j^m X_{jlt}^m - s_t^m (X_{jlt}^m - X_{jlt}^{m-1}), \quad m \in \mu : m \geq 2, l \in J : l \geq 2, t \in T \quad (4.20)$$

Constraints (4.19) and (4.20) calculate the completion time of all of the jobs on the machines $m > 1$. The values are based on the completion times of the jobs at the previous machines. The setup times are calculated in the same way as in constraints (4.17) and (4.18).

$$C_j \geq Y_l^m - M(1 - \sum_{t \in T} X_{jlt}^m), \quad j \in J, l \in J, m = \mu \quad (4.21)$$

Constraint (4.21) provides the completion time C of the job j , where M is a sufficiently large number (with a value of 10, as defined earlier). In this case, m is equal to the maximum number of machines, because the products are

only completed after processing on the final machine.

The constraint calculating the product delay (D) and the cost function are identical to the ones used for the single machine formulation ((4.8) and (4.9)).

4.4.3 Parallel machine formulation

This version of the formulation is intended to process the scenario with $m = \mu$ machines in parallel. Constraints (4.11) and (4.12) are used here to ensure that all jobs are assigned to exactly one position on only one machine, and that each position on every machine contains at most one job respectively, regardless of job type. The non-negativity constraint (4.13) is also used.

$$Y_l^m \geq \sum_{j \in J} (p_j^m + R_j) X_{jlt}^m, \quad m \in \mu, l \in J \quad (4.22)$$

Constraint (4.22) ensures that jobs cannot be completed before the associated products' release dates; however, in this case with parallel machines, a job can be completed on any machine.

$$Y_l^m \geq Y_{l-1}^m + \sum_{j \in J} p_j^m X_{jlt}^m + s_t^m (X_{jlt}^m - X_{j(l-1)t}^m), \quad m \in \mu, l \in J : l \geq 2, t \in T \quad (4.23)$$

$$Y_l^m \geq Y_{l-1}^m + \sum_{j \in J} p_j^m X_{jlt}^m - s_t^m (X_{jlt}^m - X_{j(l-1)t}^m), \quad m \in \mu, l \in J : l \geq 2, t \in T \quad (4.24)$$

Constraints (4.23) and (4.24) calculate the completion time of a job on a given machine based on machine availability, the completion time of the previous job and whether any setup time is required. The value of Y_l^m generated by these has to be larger than that generated by constraint (4.22), otherwise (4.22) takes precedence.

$$C_j \geq Y_l^m - M(1 - \sum_{t \in T} X_{jlt}^m), \quad j \in J, l \in J, m \in \mu \quad (4.25)$$

Constraint (4.25) provides the completion time C of the job j . In this case, any job can be completed on any machine. The constraint calculating the product delay (D) and the cost function are identical to the ones used for the single machine formulation ((4.8) and (4.9)).

These formulations are flexible and can be applied to a variety of situations. The modularity of this approach allows for the construction of scheduling algorithms that suit the configuration of any production line. This fills a gap in literature by providing a simple, understandable means of optimising a production schedule for products with more than one type that require sequence dependant setup times on machines. Solutions will be generated significantly more rapidly for smaller groups of machines and fewer products and product types.

Using a shifting margin for scheduling can reduce the processing time required by the scheduling formulation. An additional benefit is that a shifting margin will allow for a faster response to disruption: depending on the requirements of the factory, failed products can be scheduled for reprocessing, and unexpected machine breakdown can be scheduled around.

4.5 Example data set results and discussion

Based on the analysis of the Siemens factory HV-LC line and the requirement for a generic solution, two sets of scheduling comparison methods have been established. The first of these will use the different machine setups present in the factory production line. These are: a single machine, three machines in series, and three machines in parallel. The machines represented by these groups are defined in Section 4.3. The processing times are identical for each

machine. Two product types are used (A and B) and assigned randomly. If a machine is required to perform a job with a product of type B immediately following type A, or vice versa, it requires a period of setup time, which is equal to half the processing time. Setup time can vary significantly across processes, but can take up to 50% of the total processing time [103], so this has been chosen to represent a theoretical maximum. The setup times used can be adjusted based on industrial knowledge or the requirements of the user of the scheduler. As products would be transported around the factory in batches, they all arrive simultaneously and thus release dates are all identical and can be ignored. These batches will be identical for each comparison, with randomly assigned types. The due dates are randomly assigned and identical across the single and serial simulations. For the parallel machines, the due dates are one third of those for the single and serial machines, rounded up. Single batches of 10, 50 and 100 products will be tested, comparing the relative benefits at different scales of production. These results are presented in Tables 4.8 to 4.10.

The second comparison will use the factory setup represented in Figure 3.2. This is a simplified version of the full production line as described in Section 4.3. Despite the simplicity, it is considered representative of the original production line, and the modules used to construct it could easily be rearranged and altered to better suit other production line setups. A set of 100 products with randomly assigned types will be released to the first machine. The due dates will be randomly assigned, with a larger range than the simulations performed for the single machines. After scheduling and processing there, batches of 10 products will be released to the serial machines, as per the batch transport used in the existing factory. These products will have release dates equal to the completion time of the last product to be completed in that set, as this is the soonest that the entire batch can be transported. This process will be repeated for the parallel machines. These results are presented in Table 4.11. All of the machine setups are created and simulated using MATLAB

and Simulink. The optimisation problem is modelled using YALMIP, a toolbox for modelling and optimisation in MATLAB [160]. YALMIP is a powerful tool for modelling MILP problems and is capable of automatically selecting a suitable solver, with inbuilt interfacing for approximately 20 of the most commonly used. In this case, the MILP problem is solved with the Gurobi 9.1 solver using a student licence [161]. Gurobi is currently the best commercially available solver based on benchmark testing and has an extensive portfolio of industrial applications [161]. The solver takes advantage of heuristic methods to find initial solutions to the MILP problem, followed by the branch-and-cut method as described earlier.

The most commonly used metrics for evaluating production scheduling are work in progress (WIP) and product delays [64]. The latter is defined by the average and maximum delays, and the number of late jobs. WIP is not used in this case as it is very similar to the other metrics. The final metric to consider is the overall time taken by the software to create a schedule. The goal of the scheduling systems is to minimise these metrics, focusing on those considered most important by the users.

The average delay and the number of delayed products can be used to indicate the effectiveness of the scheduling system. The average delay is calculated by dividing the sum of the delays by the number of products. The longest delay identifies the product with the greatest difference between its due date and completion time. This can help define the distribution of the delay lengths – if the difference between the average delay and the longest delay is small, the overall variance in delay length is also small and vice versa. The total processing time for all of the products is an indicator of the length of time required to run the production line being simulated. Minimising this value would reduce the running costs of a factory. These metrics can be ranked depending upon the priorities of the manufacturer, and can be weighted to account for their relative importance. This allows them to be combined into

a single "cost function" that provides a measure of the effectiveness of the scheduling solution.

Alongside these, the final factor worth monitoring is the length of time taken by the scheduler itself. Due to the complexity of the MILP formulation, this time increases exponentially as the number of products increases. In contrast, PRBMs can quickly schedule large numbers of products, and the time taken to do so increases linearly. As a result of the time sensitive nature of industrial environments, the MILPs are left to run for a maximum of 10 minutes. The schedule created by the PRBM can also be used to initialise the MILP. This saves a significant amount of time as it allows the MILP to start searching for a solution from an existing schedule, while adding a negligible amount of time to the overall processing time. In this specific use case, the schedules are generated a day in advance while the machines are following a schedule generated the previous day, which would allow for additional processing time. However, as this is intended to be a more generic solution, this time limitation makes the results more applicable across a greater range of production lines. It makes this solution effective in situations where a schedule is required with limited advance warning.

Number of Products	10		50		100	
	PRBM	MILP	PRBM	MILP	PRBM	MILP
Average Delay	0.35	0.1	0.78	0.13	3.55	1.28
Longest Delay	1.5	0.5	3	1.5	13.5	8.5
Number of Delays	3	2	26	7	74	25
Total Processing Time	12.5	11.5	63	53.5	128	106
Cost Function	8.55	6.65	35.02	28.52	77.75	62.11

Table 4.8: Single Machine Results

The single machine results in Table 4.8 show that the MILP approach performs better than the PRMB. The cost function is reduced by approximately 20% regardless of the batch size. The reduction in the number of delayed product is also significant, ranging from 33% at 10 products to 73% at 50 products,

Number of Products	10		50		100	
	PRBM	MILP	PRBM	MILP	PRBM	MILP
Average Delay	1.75	0.9	2.28	1.5	4.39	4.39
Longest Delay	3.5	2.5	5	4	14.5	14.5
Number of Delays	9	7	43	38	87	87
Total Processing Time	14.5	13.5	65	64	129	129
Cost Function	12.55	10.65	39.02	37.06	80.75	80.75

Table 4.9: Serial Machine Results

Number of Products	10		50		100	
	PRBM	MILP	PRBM	MILP	PRBM	MILP
Average Delay	0.15	0	0.29	0	0.72	0.72
Longest Delay	0.5	0	1.5	0	3.5	3.5
Number of Delays	3	0	19	0	51	51
Total Processing Time	4.5	4	20.5	18.5	41.5	41.5
Cost Function	3	2.2	12.41	9.38	24.6	24.6

Table 4.10: Parallel Machine Results

and 66% for 100 products.

Table 4.9 shows that the increased complexity resulting from the additional machines alongside a batch of 100 products has an impact on the MILP's ability to provide a solution within the 10 minute time frame. As a result, the schedule remains the same as that generated by the PRBM. The cost function reduction is 15% for 10 products and 5% for 50 products.

As with the large batch of products on the serial machines, a batch of 100 products on 3 parallel machines does not provide an improved result in 10 minutes or under (Table 4.10). However, for batches of 10 and 50 products, all delays can be prevented, and the total processing time reduced by 11% and 10% respectively. The cost functions for these product batches are reduced by 27% and 24%.

In the factory simulation, the MILP performs better in every metric except longest delay. The cost function is reduced by 2%, and overall processing

	Production Line Simulation Results	
	PRBM	MILP
Average Delay	4.94	2.83
Longest Delay	15.5	18.5
Number of Delays	83	51
Total Processing Time	159.5	148
Cost Function	107.93	105.68

Table 4.11: Production Line Simulation Results

time by 7%. The average delay and number of delays drop by 43% and 39% respectively. The longest delay shows an increase of 1%. This is a result of the single machine sorting process. The cost function prioritises minimising the sum of the largest delay and average completion times. As a result, as long as the delay of a particular product is lower than the current maximum delay present in the system, the MILP formulation will not change it unless this will also reduce the average completion time of all of the products. Since the due dates have a larger range and higher values for this simulation, a larger portion – or all – of the products can be completed early. This means that products with earlier due dates can be placed later in the order on the single machine without becoming delayed. As such products are sent forward in their batches, this limits where they can be sorted. After processing at the serial and parallel machines, these products can end up significantly delayed. This problem could be solved by creating a series of due dates for each of the machines, which will minimise the number of products that could be affected in that way. Another approach capable of improving the results could be to tune the cost function. Depending on the requirements of the factory, the cost function can be weighted to prioritise reducing delays at the expense of the total processing time. Additionally, by associating machine processes with other costs such as energy consumption, the cost function can be altered to prioritise them rather than time and processing metrics.

4.6 Concluding Remarks

The MILP formulations presented in this chapter are capable of solving scheduling problems that include multiple product types with different machine setups. They are compared to a PRBM that represents industrial scheduling techniques. By limiting the solving time to 10 minutes, the MILP remains a viable choice for industrial applications. In addition to this, the MILP can be initialised with the values from the PRBM, which ensures that its performance is equal to or better than the standard. The reductions in the cost function range from approximately 20% for a single machine, 15-5% for serial machines, and 27-24% for parallel machines. In the scenario representing a factory environment, with batch transport of products, the MILP provides improvements to every metric except the longest delay. A potential solution to this is to provide individual sets of due dates for every stage at the factory. Despite this, the cost function is still reduced by around 2%. Alternatively, adjusting the cost function could have the same effect. Finally, associating costs such as energy consumption with the machine processes will allow the MILP to minimise them.

Chapter 5

Bayesian Change point Detection for Fault Detection and Health Monitoring

5.1 Introduction

This chapter describes offline and online non-disruptive techniques to detect a machine fault in a wave-soldering machine in Printed Circuit Board (PCB) manufacturing. The fault in question is solder build-up in the machine pipes, which was considered the most common recurring machine fault in the wave-soldering process by Siemens Congleton. The system maintains a constant flow of solder in order to provide a set wave height for PCB soldering. Over time, solder dross builds up in the pipes and reduces the wave height, leading to unsoldered products that require repairs. Once this is detected, pump power is manually increased to raise the wave height, compensating for the reduced pressure. After the power is raised over a threshold, the machine is turned off and maintenance is performed. This is a reactive process repeated on a weekly to monthly basis, depending on the length of machine up-time and quality of solder. The wave-soldering machine has to be shut down during this time,

for several hours up to several days depending on the severity of the fault. The goal of the presented techniques is to be low-cost and non-disruptive, while improving the fault detection rate and reducing the number of released unsoldered products.

Both techniques presented in this chapter use Bayesian Change point Detection (BCD). Pump vibrations are recorded as solder build up occurs. The offline fault-detection method retroactively analyses a full set of these vibrations gathered over a whole maintenance cycle. It can isolate the pump responses that are indicative of a machine fault as well as informing users of the expected machine up-time before faults occur.

The online fault-detection method analyses incoming data in real time. Over a maintenance cycle, as new pump data is recorded, it is compared to the previously collected data and the likelihood of a change point having occurred is assessed. This information can be passed on to the machine operators in real time, presenting the appearance of a fault as the probability of a change in machine state.

5.2 Bayesian Change point Detection Background

5.2.1 Overview of Bayesian Statistics

Bayesian statistics is an approach to data analysis and parameter estimation based on Bayes' theorem (equation 5.1) [132], [162]. It uses observed data to update the available knowledge about the parameters in a statistical model, and treats unknown parameters as random variables with known prior distributions [163]. Bayes' theorem revolves around the events A and B , with A usually representing an unknown or unobservable quantity or parameter of interest, and B the evidence, or newly acquired data. It is composed of the functions outlined in the following paragraphs.

$P(A)$ is the *prior distribution* of A and the probability of observing the event without any given condition. A prior distribution is defined by the available knowledge about the parameters, and is typically determined before data collection. Priors are generally classified as diffuse, weakly informative, or informative, reflecting the certainty or knowledge about the model parameters being estimated [132]. Informative priors reflect high degrees of certainty or knowledge about the parameters. Weakly informative priors incorporate some amount of knowledge about the parameters but are used when there is less certainty about the values. These have a relatively smaller impact on the posterior, which would be weighted more by the data observations as expressed in the likelihood. Diffuse priors indicate complete uncertainty, often just used as placeholder before analyses are conducted with more informative priors. Improper priors have distributions that integrate to infinity and can lead to an improper posterior: this impacts the degree to which the results can be substantively interpreted.

$P(B)$, the probability of the evidence, is calculable by summing or integrating

across all values of A , weighted by how strongly we believe in each particular value of A (where A is the parameter). It is often omitted, as it can be difficult to calculate and it does not change during a particular analysis. In these cases, it is treated as a constant/normalising factor. Additionally, it can be used to assess the model and how well it fits the data. Notably, $P(B) \neq 0$.

$P(B | A)$ is the *likelihood function*, the probability that B would be observed, given that A is true. It quantifies the extent to which the evidence B supports the proposition A , for a given choice of A where A is the model parameter. This indicates the compatibility of the evidence with the given hypothesis.

The term $P(A | B)$ is the conditional probability of A given that B is true and is referred to as the *posterior probability*. This is the quantity of interest: combining both the prior distributions and likelihood function to generate updated parameters based on observed data.

These functions can be combined to state Bayes' theorem mathematically as follows:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad (5.1)$$

The structure of a Bayesian data analysis problem can be expressed most simply with the following steps [132],[162]:

1. Setup of the full probability model: defining the probability distributions for the observable and unobservable quantities in the problem. This model should reflect the current knowledge about the problem or system being assessed.
2. Conditioning on the observed data: calculating and interpreting the appropriate posterior distribution, the conditional probability of the unobserved parameters of interest given the observed data.
3. Evaluation of the fit of the model and the implications of the resultant

posterior distribution: assessing the fit of the model, the conclusions that can be drawn from the results and the sensitivity of the model.

The final step can either be the end of the process, or an opportunity to refine or expand the model and repeat the process. In particular, it is a chance to assess the impact of the chosen priors and decide whether they need to be updated to better reflect their effect on the overall model.

Fitting a model to the data to obtain the posterior distribution (step 2 of a Bayesian problem) can be done in a number of ways, both analytical and numerical. Analytical solutions tend to be lengthy and complex, and can become intractable as the number of priors and the complexity of the problem increases [132], [162], [164]. For this reason, it is often the case that the posterior distribution cannot be evaluated exactly.

To overcome this, a number of methods have been developed to estimate the posterior distribution. The most common of these are Markov Chain Monte Carlo (MCMC) methods, which combine Markov chain processes and Monte Carlo algorithms [133]. While an overview of MCMC methods was provided in Chapter 2, more detail will be presented in the following paragraphs to provide more context and detail for this chapter. Generally speaking, Monte Carlo algorithms are techniques for estimating integrals from randomly generated values sampled from a given distribution. In this case, this means a Monte Carlo integration can obtain a distributional estimate of the posterior distribution and its associated statistics from a set of samples from that distribution. However, since it is impossible to directly and independently sample parameter values from the posterior distribution, Markov chains are used. A Markov chain generates a sequence of values, which, at any given point, are dependant only on the immediately preceding value. By using a specified transition kernel, which defines the updating procedure of the parameter values within a Markov chain, the Markov chain is capable of reaching a stationary distribution of values equal to the posterior distribution of interest. These values can

be treated as samples from the posterior distribution by the Monte Carlo algorithm, which can then express that posterior distribution for interpretation by the user.

Markov chains are defined by the specification of the initial parameter values and transition kernel. These, in turn, are defined by the chosen approach, which intends to ensure that the generated stationary distribution is the correct posterior distribution [133]. A number of the most common approaches will be outlined here [132]. One of the first to be developed was the Metropolis-Hastings algorithm, which uses a general proposal distribution with an associated accept/reject step for the proposed parameter value(s) [165], [166]. The Reversible Jump MCMC algorithm extends Metropolis-Hastings and permits the simulation of trans-dimensional jumps within parameter space [167], [168]. Hamiltonian Monte Carlo is a Metropolis-Hasting algorithm that uses Hamiltonian dynamics in situations where direct sampling is difficult, if the sample size is small or if auto-correlation is high [169]. The No-U-Turn Sampler (NUTS) is an extension of Hamiltonian Monte Carlo that is often faster because it can avoid the need for tuning the model [170]. The Gibbs sampler is another Metropolis-Hastings algorithm where the proposal distribution is the corresponding posterior conditional distribution, with an associated acceptance probability of 1 [171]. Particle MCMC uses a combination of Sequential Monte Carlo (outlined below) and MCMC for problems with analytically intractable likelihood functions [172]. Finally, Evolutionary Monte Carlo incorporates features of genetic algorithms and simulated annealing, allowing the Markov chain to effectively and efficiently explore the parameter space and avoid getting trapped at local modes of the posterior distribution [173]. The choice of approach is dependant upon the requirements of the problem being solved.

A non-exhaustive list of less common methods for model fitting through sampling are outlined here for comparison. Sequential Monte Carlo (SMC) uses

multiple importance sampling steps for each observed data point, and is particularly valuable for online data analysis [174]. Approximate Bayesian Computation (ABC) is an approximate approach typically used when the likelihood function is analytically intractable or particularly computationally expensive [175]. Integrated nested Laplace approximations is an approach developed for a specific type of model: latent Gaussian models, which include generalised additive spline models, Gaussian Markov processes and random fields [176]. Variational Bayes approximates posterior distributions via simpler approximating distributions, which can then be optimised to minimise a loss function known as the evidence lower bound [177]. Other methods can be found collected in the literature [178].

5.2.2 Bayesian Change point Detection

Offline Change point Detection

Change Point Detection (CPD) is used to find the number and location of any change points in a data set, where a change point is defined as a change in the probability distribution of the measured data. When performed offline, this occurs retroactively, splitting existing data sets into subsections based on their expected state. This is done by selecting a parameter of the data - for example, the mean of the collected samples. Changes to this mean outside of a range or a trend are considered indicative of a change point. The use of Bayesian inference as part of Bayesian Change point Detection (BCD) allows the occurrence of a change point to be presented as a probability based solely on the available data, while also indicating the size of the change as well. The methodology for finding a change point in a data set is presented below, following examples of numerical implementations found in the literature [138], [179].

The initial step requires the definition of a model for the distribution of the

collected data. The data can be classified as discrete, continuous or mixed and the chosen distribution has to reflect that. For this example, it is assumed that the data is collected continuously over a period of time and can be represented by a stationary Gaussian (or Normal) distribution $N(\mu, \sigma)$ with a mean value μ and a variance σ^2 . Assuming that there are J observations, the collected data points can be given by $\mathbf{d} = \{d_1, \dots, d_J\}_{i=1, \dots, J}$. Finally, the number of expected change points and the expected effect of a change point is defined. In this case, which assumes a single change point, which indicates a change in the mean value of the data set but not the variance, gives the following:

$$d_i = \begin{cases} N(\mu_1, \sigma) & \text{for } t \leq \tau \\ N(\mu_2, \sigma) & \text{for } t \geq \tau \end{cases} \quad i \in 1, \dots, J \quad (5.2)$$

where d_i is any data point, τ is the change point and t is the time at which the data point d_i is observed. As only the observed data is known, but no information about its distribution is known, the change point and its probability distribution have to be inferred purely from the measurements. This can be expressed using Bayes' theorem:

$$P(\mu_1, \mu_2, \sigma, \tau \mid d) = \frac{P(d \mid \mu_1, \mu_2, \sigma, \tau) P(\mu_1, \mu_2, \sigma, \tau)}{P(d)} \quad (5.3)$$

Equation (5.3) can be solved analytically [164] by marginalising it with respect to μ_1 , μ_2 and σ . This provides an analytical solution to the posterior probability distribution $P(\tau \mid d)$. As discussed above, the posterior probability is a model parameter inferred and updated by the observed data - in this case, it represents the inferred probability that a change point τ occurred at a given data point d_i . However, in many practical cases, analytical solutions can become mathematically intractable [138] and as such numerical methods are employed instead. This requires prior distributions to be assigned to the unknown parameters $P(\mu_1)$, $P(\mu_2)$, $P(\sigma)$ and $P(\tau)$, using a best guess based on knowledge

about the system or process being studied. The level of confidence in this knowledge will determine the informativeness of the prior distributions, from informative to diffuse, and care should be taken at this stage to ensure the chosen priors are not improper.

Once prior distributions have been assigned, this problem is solvable using readily available tools, as discussed in the previous section. In this case, PyMC3 is used because of its flexibility and power [138] [180]. It provides access to a number of MCMC methods that can sample from the given prior distributions in order to generate the posterior distribution $P(\tau|d)$. PyMC3 can automatically take advantage of multiple MCMC chains to improve the quality of the results [181] and discards the first set of samples that form the "burn-in period" which MCMC methods experience before convergence.

Online Change point Detection

While MCMC is the favoured solution to static Bayesian change point detection problems, it is relatively computationally expensive and would have to re-analyse the entire data set as new data points are collected. For this reason, sequential change point algorithms are used, which are capable of handling a constant stream of new data and making a decision about the occurrence of a change point based only on the data observed to that point [134]. This forms the basis of online Bayesian Change point Detection (OBCD), which was initially proposed simultaneously by Adams and MacKay [135] and Fearnhead and Liu [136]. The former uses a recursive message passing algorithm to calculate the probability distribution of the current "run length", the distance since the last change point. Each data set is assumed independent of all others, as per [182], and an underlying predictive model and hazard function define the likelihood of a change point at a given run length. The latter proposes an online algorithm for the exact filtering of multiple change point problems, enabling direct simulation from the true joint posterior of the number and po-

sition of change points. Both suffer from a rapidly increasing computational cost over time as the number of samples increases. A solution is presented in the form of a maximum sample size, such that old samples are discarded as new ones come in. An evaluation of change point detection algorithms found that Adams and MacKay's algorithm performs favourably when compared to other existing algorithms [183].

A large portion of existing sequential Bayesian change point algorithms, both those based on the methods highlighted above and otherwise, impose a prior distribution on "run length", which can strongly impact their inference. It is also considered an unrealistic assumption that the frequency of change point occurrence is fixed and known in advance [184]. To overcome these issues, this chapter focuses on the work done by [134], which presents a sequential Bayesian change point algorithm that allows direct simulation from the exact posterior distribution on change point locations without imposing any assumptions on the distance between adjacent change points. This algorithm also maintains a linear time complexity per time step by storing and adding to previously calculated distributions rather than rerunning the entire analysis for each new data point. The formulation of the algorithm is outlined below.

The first step is to make an initial assumption about the data model: for this example, a linear regression model is used to predict the distribution of the data. However, it is important to note that a wide range of underlying predictive models can be used, limited only by the user's ability to specify an appropriate model.

$$Y = \sum_{l=1}^m \beta_l X_l + \varepsilon \quad (5.4)$$

Equation (5.4) presents the linear regression model used to represent the data gathered, where Y is a dependant variable with m known predictor variables $X_1, \dots, X_m$. β_l is the l th regression coefficient and ε is the random error term.

The assumptions made about β and ε are the same as in [134], with the same conjugate priors assigned here for both.

$$f(Y_{i:j}) = f(Y_{i:j}|X) \quad (5.5)$$

Assuming that this regression model applies to the data between the start of the data set up until the change point, and from the change point to the end of the data set, for each possible subset of the data $i : j$, the probability of the data given the regression model is calculated using equation (5.5) and stored.

$$P_1(Y_{1:j}) = \sum_{n=1}^{j-1} f(Y_{1:n})f(Y_{n+1:j}) \quad (5.6)$$

In equation (5.6), the probability density of the first j data points containing the change point is calculated and stored. Starting at the beginning of the time series, two non-overlapping subsets of data (assumed to be independent as per [182]) are pieced together. By multiplying together the probabilities of the two subsets (calculated in equation (5.5)) and summing over all of the potential locations of the change point, the probability density of the data can be calculated.

The final quantity that needs to be specified is the prior distribution of the location of the change point. By using an uninformative prior distribution which assumes that the location of the change point is equally likely at any data point, bias in the algorithm can be avoided. From here, sampling from within the distributions defined above can be performed to ascertain the location of the change point and the regression parameters for the subsets of data either side of it.

This is extended in [134] to search for additional change points up to a defined maximum number of change points k_{max} . This is outside of the scope of this chapter, but is achieved by extending equation (5.6) such that the term $f(Y_{1:n})$

contains $k - 1$ change points when calculating the probability of the location of the k th change point. The prior distribution for the number of change points is specified, and sampling for change points between 0 and k_{max} takes place before sampling for the locations of the change points. Additional tuning can be done based on the values of the linear regression model used, as described in [134], [139]. A minimum distance between change points can also be specified. The described algorithm will locate a change point in a single set of data of a set length [185]. To expand it efficiently to process incoming data points, the matrices storing the results of equations (5.5) and (5.6) are expanded. New rows and columns are added to hold the new probabilities calculated for each newly generated subset of data (by equation (5.5)). A new column is added for the new probability density $P_1(F_{1:N})$ where N is the most recently added data point. The full implementation details for multiple change points can be found in [134].

5.3 Test bench and data collection

The test bench used to formulate and test the BCD method is presented in Chapter 3 and considers the behaviour of a pump moving water through a closed system. It represents the movement of solder through the wave-soldering machine as pipe blockage takes place over time. An accelerometer attached to the pump records its vibrations as a gate valve is shut to mimic a pipe blockage. A flow meter records the corresponding flow rate as the valve shuts to assess the results of the analyses.

Other test bench setups were considered, including a variant that created a vertical wave in the water tank whose height could be monitored. The pump generates a turbulent flow, leading to a turbulent wave. This mimics the chip wave that PCBs encounter first as they are carried across through the wave-soldering machines. However, accurate measurement of the top of the wave

proved difficult as a result of the turbulence. Assuming that all other factors are kept identical, the wave height is controlled by the flow rate of the fluid. For this reason, the presented test bench is the one used to collect data.

A fast Fourier transform (FFT) [118] is used to extract the frequency and amplitude data from the vibrations of the pump as recorded by the accelerometer. Figures 5.1 and 5.2 show the results of the FFT on the vibration data recorded from an open and shut valve respectively. The difference between the two is clearly visible, as the peak frequency shifts to the left. A lot of noise is present in the system – as would be present in an industrial setting. A moving average filter is used as it can, quickly and with low computational cost, reduce the noise in a signal [186], and doubles as a low-pass filter to mitigate the possible effects of aliasing. The solder in the use case might result in different vibrational characteristics. Despite this, the solder blockage should still result in a change in the recorded vibrations over time, which can be detected by the BCD algorithm. This is a significant benefit of the proposed methodology: as long as the input changes in a way that can be detected, the BCD algorithm will be able to assess the probability that a change point has occurred at a given point in time. This forgoes the need for calibration procedures, training data and behavioural models, which are unavailable in this scenario and can be expensive and time consuming to generate regardless.

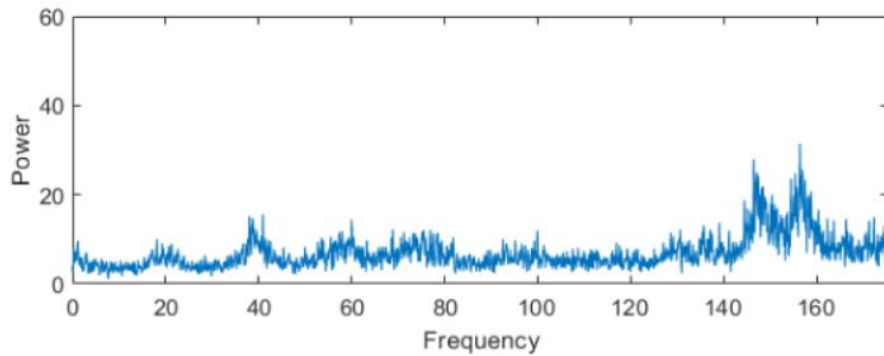


Figure 5.1: *FFT of pump vibrations while the valve is open*

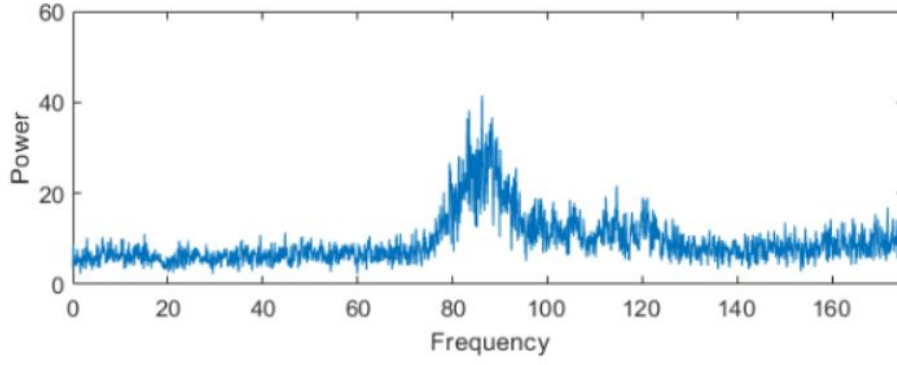


Figure 5.2: *FFT of the pump vibrations while the valve is closed*

5.4 Offline BCD

5.4.1 Formulation

Figure 5.3 presents the pump data gathered by the accelerometer. For this chapter data was collected as the gate valve was closed in a number of discrete increments, going from 0 (completely open) to 25 (completely closed) . Analysis of the time-domain vibration data has shown that it is not suitable for the analysis – it does not vary significantly enough across the different valve states to be a useful feature for predicting the current state. The mean of the amplitude of the time-domain data is shown to remain roughly constant throughout. However, the plotted features extracted from the FFT (peak frequency and maximum amplitude) can be seen to respond to the change in valve state. As such, these are the features used in the BCD algorithm presented in this chapter.

Since both the peak frequency and the maximum amplitudes are used for the BCD in an identical fashion, the formulation is only shown for the maximum amplitude and will follow the implementation outlined in the Offline Change point Detection subsection above. In this case, the amplitude (A) of the vibration data is continuous and samples are collected over the range of valve states, V , such that $A = \{A_1, \dots, A_V\}_{v=1, \dots, V}$. It is safe to assume that any given sample for A will have a value that falls within a normal distribution:

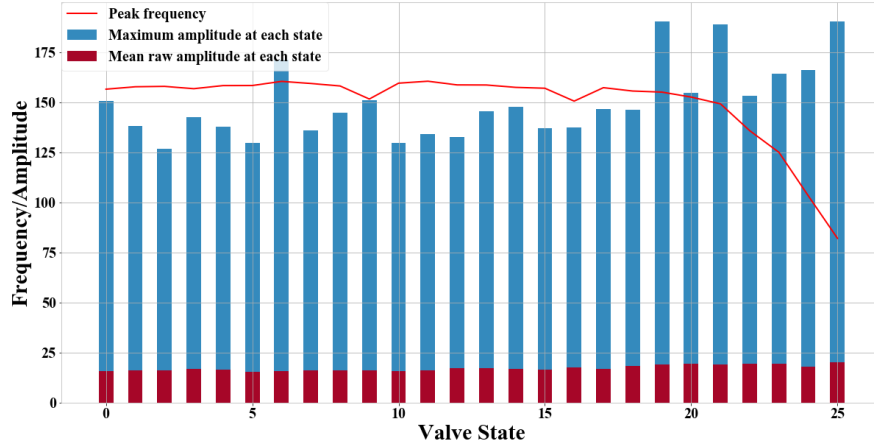


Figure 5.3: *Data Extraction from Recorded Pump Vibrations*

$$A_V \sim N(\mu, \sigma) \quad (5.7)$$

It is then assumed that over the range of valve states there is a single change point, τ , during which μ changes to a different value – corresponding to the point where the mean amplitude changes and the pipe state deteriorates enough to significantly affect the flow rate. This leads to two μ parameters, one before τ and one for the rest of the remaining observation period:

$$A_V = \begin{cases} N(\mu_1, \sigma) & \text{for } v < \tau \\ N(\mu_2, \sigma) & \text{for } v \geq \tau \end{cases} \quad (5.8)$$

Since only the observed data is known, but no information about its distribution is available, the change point and probability distribution have to be inferred purely from the measurements. This can be expressed using Bayes' theorem:

$$P(\mu_1, \mu_2, \sigma, \tau | A) = \frac{P(A | \mu_1, \mu_2, \sigma, \tau) P(\mu_1, \mu_2, \sigma, \tau)}{P(A)} \quad (5.9)$$

Prior distributions are now assigned to the unknown parameters $P(\mu_1)$, $P(\mu_2)$,

$P(\sigma)$, and $P(\tau)$. Based on the observed data, the following prior distributions are assigned: $\mu_1 \sim \text{Uniform}(100, 200)$, $\mu_2 \sim \text{Uniform}(100, 200)$, $\sigma \sim \text{Uniform}(0, 100)$, and $\tau \sim \text{Discrete, Uniform}(0, 25)$. Since the actual values for μ_1 , μ_2 and σ are unknown, their prior distributions are assumed to be uniform over a range informed by the observed data. The value of τ is also unknown, but is known to be discrete as it has to be one of the tested valve states. Uniform distributions have been assigned to all the parameters to avoid biasing the BCD. However, other distributions can be used based on knowledge about the system in question.

Once the distributions have been assigned, this problem is solvable using readily available tools: in this case, PyMC3 is used because of its flexibility and power [138], [180]. It uses MCMC methods, which generate samples from the given prior distributions and compares them to the existing data in order to generate the posterior distributions. In this case, the method used is adaptive slice sampling [181], chosen because of its suitability for sampling from discrete data distributions. Multiple MCMC chains are used to improve the quality of the results [187], with the first set of samples being discarded as part of the “burn-in period” which occurs before convergence. With this done, the distribution of τ , μ_1 , μ_2 and σ can be plotted with simple functions of PyMC3. The plot of τ shows the likelihood of the change point occurring at any given valve state.

5.4.2 Results and discussion

From Figure 3.9, it can be seen that flow rate reduction because of pressure drop is minimal until valve state 18 – approximately 75% closed. After this point, the flow rate reduction appears to be linear.

Figure 5.4 shows the plotted distributions of τ , μ_1 , μ_2 and σ when applying the algorithm to the extracted maximum amplitudes. The higher value of

μ_2 over μ_1 confirms that a change point has occurred and the amplitude of the pump vibrations increased as the valve was shut. The lack of an overlap between the values increases the certainty in the result [138]. The τ plot shows that the probability of the change point occurring is highest at valve state 19, although it does indicate a possibility of a change point as early as state 17. This corresponds with the flow rate plot in Figure 3.9. This information can then be plotted back onto the original amplitude reading to show the expected values, shown in red, based on the change point probabilities (Figure 5.6). The cumulative sum of τ is used to calculate the expected values: while $\sum_{V=1}^{V=v} \tau = 0$, the expected value is μ_1 , and when $\sum_{V=1}^{V=v} \tau = 1$, the expected value is μ_2 . As τ is a distribution over several valve states, the change in the cumulative sum of τ is reflected as a proportional change in the expected value from μ_1 to μ_2 .

Performing a similar analysis using the peak frequency gives the results plotted in Figure 5.5. In this case μ_1 is greater than μ_2 as the frequency drops as the valve closes. τ indicates the change point is most likely to have occurred at valve state 23, with lower chances of it being at state 22 and 24. Calculating the expected values and plotting them against the original frequency readings shows that they follow the frequency change very closely (Figure 5.7).

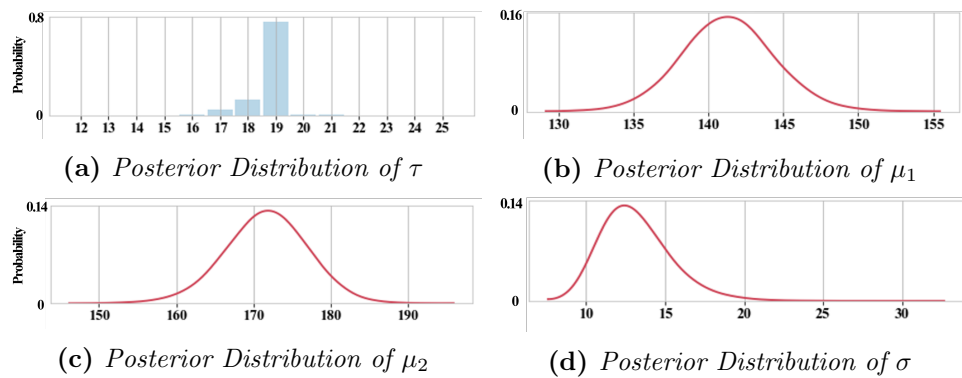


Figure 5.4: Posterior Distribution of Amplitude Parameters

This data analysis has shown that it is possible to use pump vibrations to assess the state of a system. It is possible to generate two different sources of data from a single vibration sample, which can then be used by the BCD algorithm

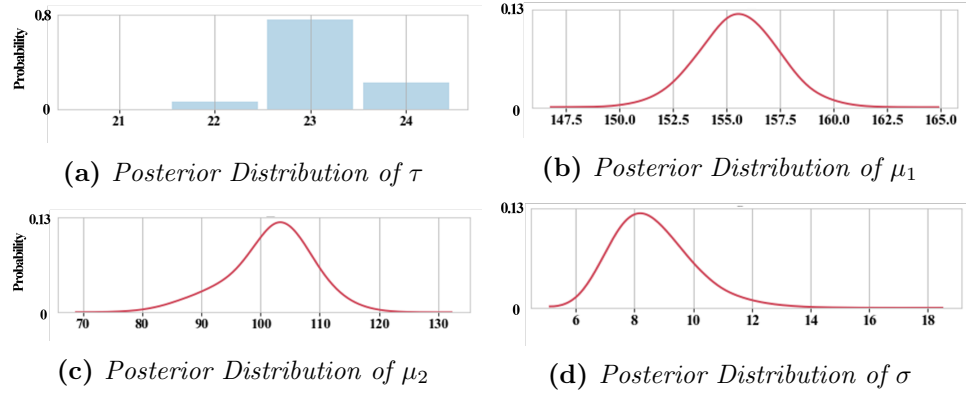


Figure 5.5: Posterior Distribution of Frequency Parameters

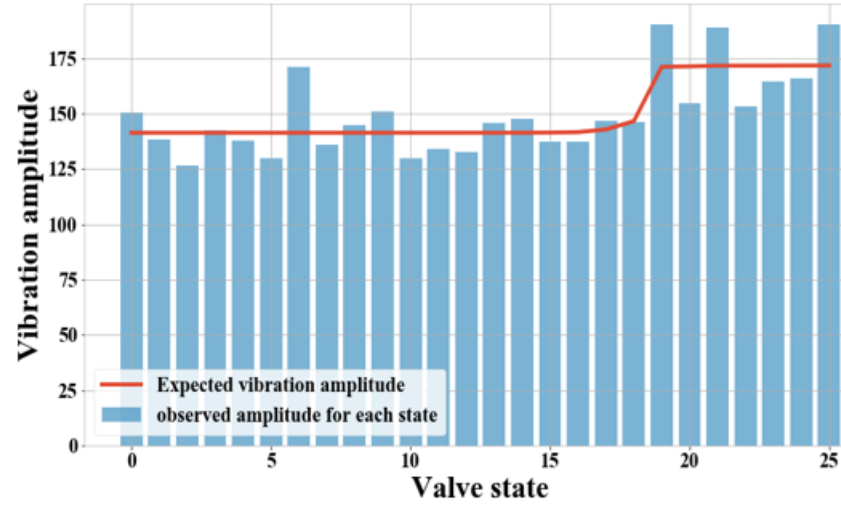


Figure 5.6: Expected Amplitude Values

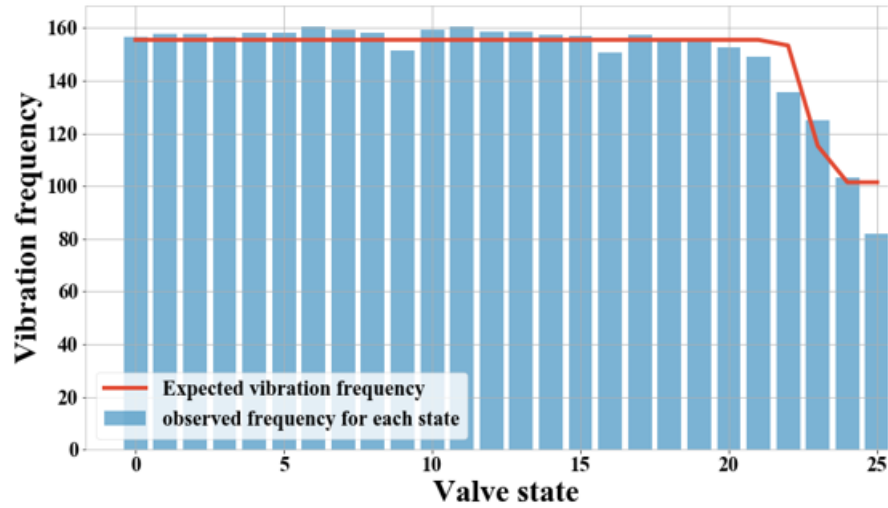


Figure 5.7: Expected Frequency Values

to retroactively identify the points at which the flow rate begins to decrease. Furthermore, this is done with a very small sample size. The predicted change

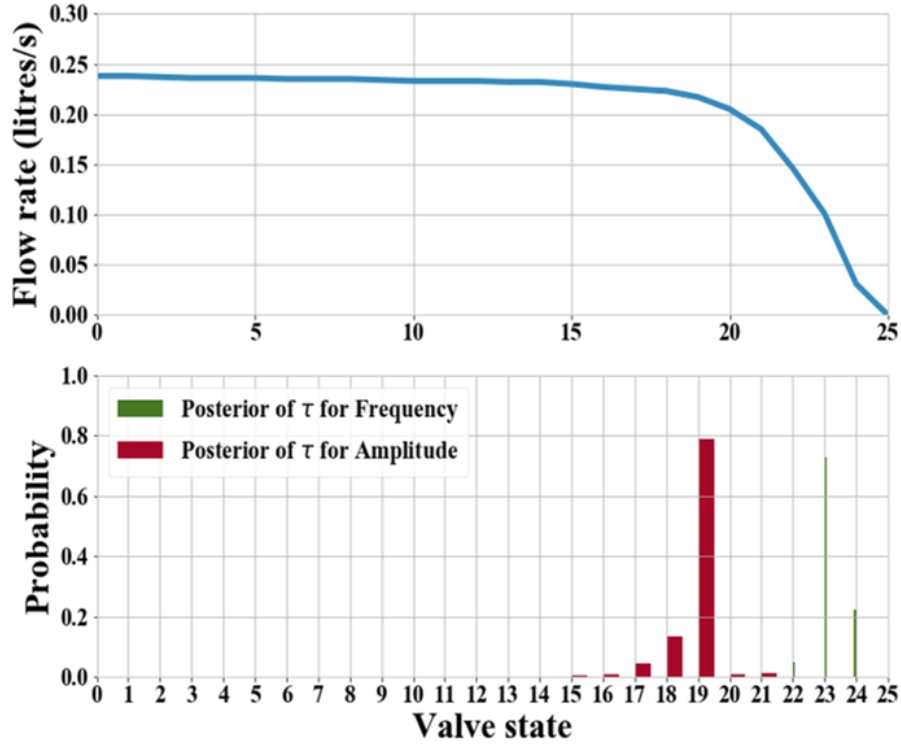


Figure 5.8: *Flow Rate and Predicted change points Against Valve State*

points can be plotted against the recorded flow rate to demonstrate the effectiveness of the BCD based on the information it used (Figure 5.8). Analysis using the amplitude leads to earlier detection, which is valuable in situations where the processes involved are sensitive to any change. Analysis using the frequency leads to later detection, useful in processes which are less sensitive to changes and do not require immediate maintenance. In this case, the wave soldering machine chosen as a use case is sensitive to rapid changes in flow rate, so earlier detection is desirable.

This fault detection would be performed using the extracted μ values for the amplitude and frequency of the pump vibrations. As the pump behaviour was effectively segmented by the change point into healthy and faulty behaviour, it can be assumed that an amplitude or frequency reading of μ_1 would indicate healthy behaviour. With this knowledge, pump vibrations could be monitored live and a user notified if the extracted amplitude or frequency fell outside of the distribution presented in Figures 5.4 and 5.5. Larger deviations from the

expected value can be flagged as requiring more urgent attention. Depending on the rate of data acquisition, a moving average can be used to smooth out the fluctuations in the readings. As the readings approach μ_2 , it can be assumed with greater confidence that the pump behaviour is trending towards faultiness. The values of μ_1 and μ_2 can be refined over time. Additionally, a value of μ that fell outside of the ranges of μ_1 and μ_2 could effectively identify any anomalous behaviour. Finally, the length of time taken to reach the change point can be interpreted as the length of machine up-time during which behaviour can be assumed to stay healthy. This assumes that the deterioration of the machine is proportional to machine up-time, which is the case for the wave soldering machine use case in question.

The offline BCD analysis offers a number of advantages over comparable methods. Despite the retroactive nature of the analysis, the extracted features can be used to assess machine health in real time. The pump responds to the transition from healthy to faulty behaviour as it happens, leading to a response that can take place during said transition at the earliest. As the flow rate drop does not accelerate immediately, rapid response on the part of the user can prevent or reduce the production of unsoldered PCBs. It offers a number of benefits over comparable methods: for example, the use an analytical model that captures expected behaviour for comparison with observed behaviour can similarly detect faults as they take place [120] but requires significant knowledge of the system in order to build a sufficiently detailed model that accurately represents the system's behaviour. This knowledge can be time consuming and expensive to obtain, especially in cases where system experts are not available [88]. Similarly, machine learning techniques, such as the use of an Artificial Neural Network (ANN) [121], require well labelled training data that captures both healthy behaviour and all possible variations of the faulty behaviour to be detected. Normal blockage detection techniques are significantly affected by the locations of the sensors relative to the blockage

[128], another issue that this method circumvents as a result of the requirement to indirectly monitor the pipes due to the harsh working environment. Furthermore, this particular application of offline BCD analysis demonstrates its capability to analyse small sample sizes, a benefit of the use of Bayesian inference [131]. This demonstrates that initial results can be collected even with a limited data set. The drawback is the reduced precision of the results: for example, the location of the change point can only be stated with relation to the number of valve states, each of which represent a change of approximately 5% in the pipe clearance. This is likely to be partially responsible for the later detection of a change point by the peak frequency analysis. However, the small sample sizes do allow for quick initial analyses to take place before more involved investigations are performed. Larger samples sizes can be processed to increase the precision of the results (in the form of a more accurate change point location distribution) at the cost of increased processing time. The BCD algorithm is also data agnostic, which is particularly suitable for the intended future application across the machine fleet.

5.5 Online BCD

5.5.1 Formulation

This chapter uses an extended version of the data set presented for the offline BCD, as shown in Figure 5.9. A total of 250 samples have been taken as the valve goes from fully open to fully closed. The same features are used in the online analysis (peak frequency and maximum amplitude) as the mean amplitude doesn't vary significantly as the valve closes. The application of the algorithm is further modified such that the complete data set can be submitted and it will perform the OBCD process one data point at a time, to mimic incoming live measurements.

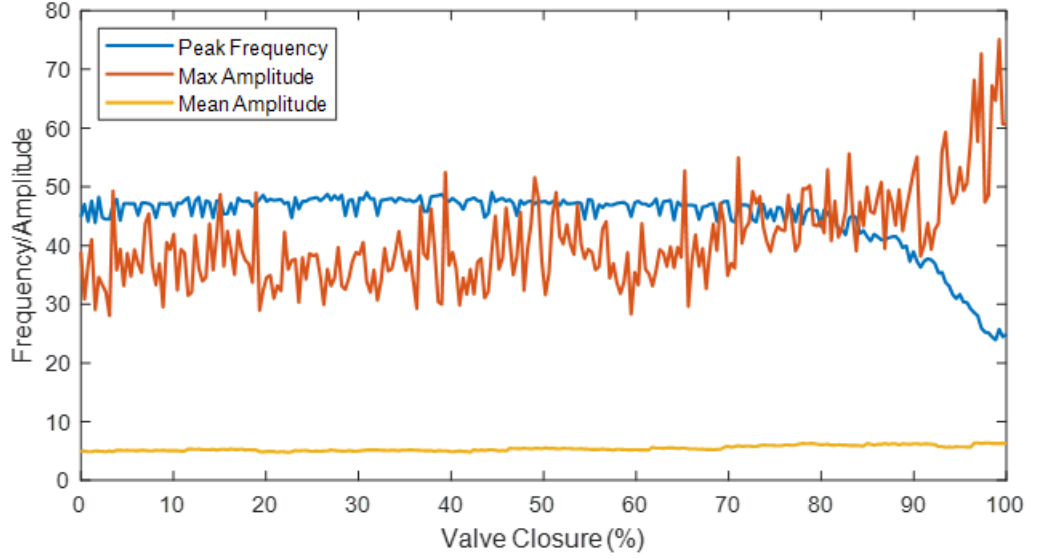


Figure 5.9: *Data Extraction from Recorded Pump Vibrations (Extended)*

The algorithm is based on the modified version of the sequential BCD algorithm described in [134], and adapted as shown earlier in the chapter to detect a single change point in the data sets used. The modified algorithm used is presented below, with the variable for the data set of interest substituted in place of the generic variables presented earlier. As the algorithm processes the peak frequencies and maximum amplitudes in the same way, the following example will be presented for only the peak frequency. An initial assumption is made about the data model: for this example, a linear regression model is used to predict the distribution of the data. Both a linear and constant model are used in the final application, for the frequency and amplitude respectively.

$$F = \sum_{l=1}^m \beta_l V_l + \varepsilon \quad (5.10)$$

Equation (5.10) presents the linear regression model used to represent the data gathered, where F is the frequency of the pump vibrations and V_l is the level of valve closure. β_l is the l th regression coefficient and ε is the random error term. The assumptions made about β and ε are the same as in [134], with the same conjugate priors assigned here for both.

$$f(F_{i:j}) = f(F_{i:j}|V) \quad (5.11)$$

Assuming that this regression model applies to the data between the start of the data set up until the change point, and from the change point to the end of the data set, for each possible subset of the data $i : j$, the probability of the data given the regression model is calculated using equation (5.11) and stored.

$$P_1(F_{1:j}) = \sum_{n=1}^{j-1} f(F_{1:n})f(F_{n+1:j}) \quad (5.12)$$

In equation (5.12), the probability density of the first j data points containing the change point is calculated and stored. Starting at the beginning of the time series, two non-overlapping subsets of data (assumed to be independent as per [182]) are pieced together. By multiplying together the probabilities of the two subsets (calculated in (5.11)) and summing over all of the placements of the change point, the probability density of the data can be calculated.

The final quantity that needs to be specified is the prior distribution of the location of the change point. By using an uninformative prior distribution which assumes that the location of the change point is equally likely at any data point, bias in the algorithm can be avoided. From here, sampling from within the distributions defined above can be performed to ascertain the location of the change point and the regression parameters for the subsets of data either side of it.

The described algorithm will locate a change point in a single set of data of a set length. To expand it efficiently to process incoming data points, the matrices storing the results of equations (5.5) and (5.6) are expanded. New rows and columns are added to hold the new probabilities calculated for each newly generated subset of data (by equation (5.5)). A new column is added for the new probability density $P_1(F_{1:N})$ where N is the most recently added

data point. The full implementation details can be found in [134].

5.5.2 Results and Discussion

Figure 3.9 shows that the flow rate reduction as a result of the valve closure is minimal until the valve is approximately 70% shut. A short transition period can then be seen until the valve is about 80% shut. The flow rate reduction from that point on is nearly linear until it reaches 0 litres/s. During the experiment, a total of 260 pump vibration samples were collected as the valve went from completely open to completely closed. This version of the algorithm was run on MATLAB, but has also been translated to Python.

Fig. 8 plots the extracted amplitudes against the resultant posterior probability of the change point's location as calculated by the algorithm. It suggests that a change point has taken place as the valve is 70% closed and shows a high certainty over a small area. This corresponds with the start of the transitional flow rate period in Figure 3.9. A plot of the linear model inferred by the algorithm can also be seen in green in Figure 5.10. It follows the collected data quite closely up until the last few readings. Had there been additional samples collected afterwards, the algorithm would have adjusted the inferred model to better match the data.

Performing the same analysis on the peak frequencies gives the results plotted in Figure 5.11. In this case, the change point is detected later, when the valve is approximated 80% closed, and the posterior probability is lower and spread over a larger area. It corresponds more closely with the end of the transitional period of flow rate noted in Figure 3.9. The inferred model follows the extracted frequencies very closely.

Plotting the posterior probabilities of the change point as calculated from the amplitude and frequencies extracted from the pump vibrations against the flow rate produced by the pump as the valve is closed produces Figure 5.12. By

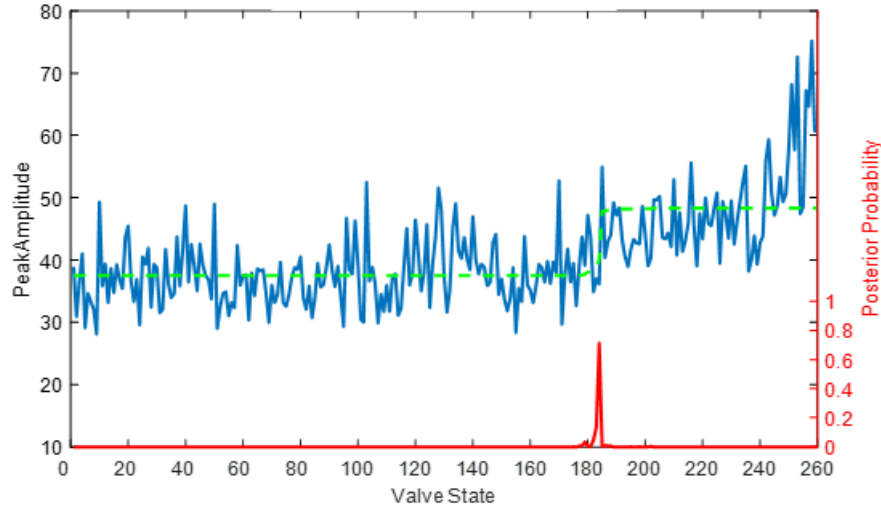


Figure 5.10: Predicted change point based on Extracted Amplitudes. The inferred linear model is represented in green.

applying this algorithm to the two different sources of data extracted from the pump vibrations, it has been demonstrated that the state of this system can be assessed by the vibrational response of the pump. Using the amplitude leads to an earlier detection of a change point, which is valuable in processes sensitive to changes in the monitored parameters. The frequency leads to a later detection of the change point, more valuable in processes that do not require immediate maintenance or are less sensitive to changes. The wave soldering machine use case presented in this chapter falls into the former category and benefits from early detection of a change in flow rate so that maintenance can be performed before faulty products are released.

Similarly to the offline analysis, the OBCD algorithm allows for the detection of a change point during the transition between healthy and unhealthy behaviour. This means that a prompt response can significantly reduce the number of unsoldered products that require repairs. However, it should be noted that it is difficult to assess the exact response time of the OBCD to a change point, and further analysis might be necessary to establish how quickly the algorithm can notify a user and allow them to take the relevant action. Despite this, the OBCD algorithm functions without the use of a supplementary analytical

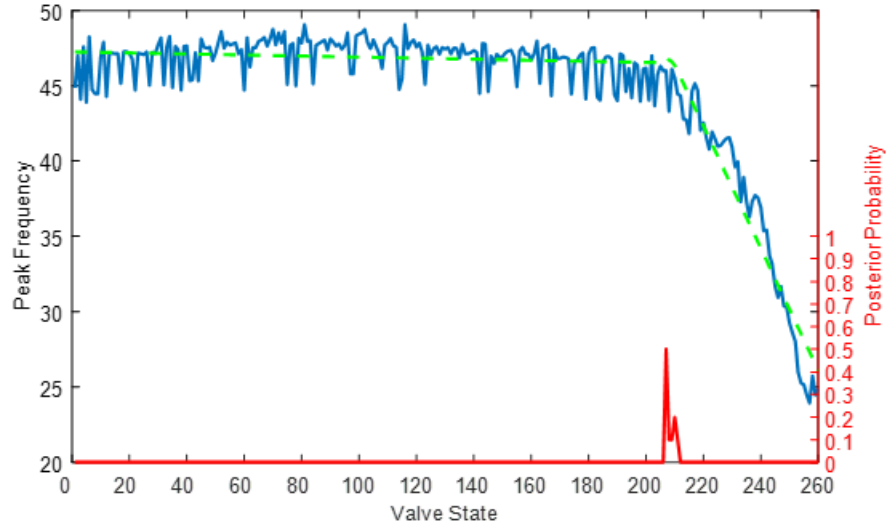


Figure 5.11: Predicted change point based on Extracted Frequencies. The inferred linear model is represented in green.

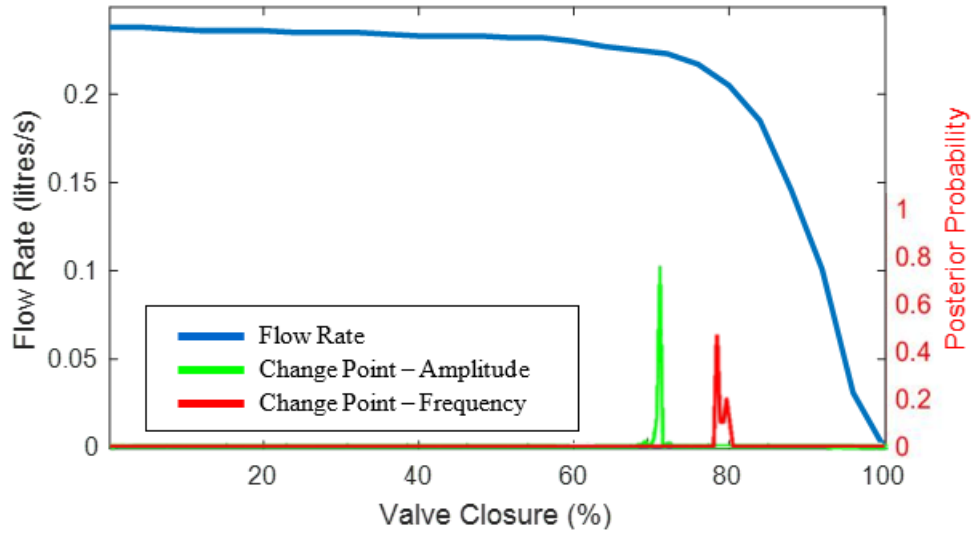


Figure 5.12: Predicted change points plotted against the Flow Rate

model [120] or labelled training data sets [121], and is unaffected by location of the blockage relative to the sensor [128]. Additionally, like the offline BCD method, the data agnostic nature of OBCD is a significant benefit and suits the intended generic application across a machine fleet.

5.6 Concluding Remarks

This chapter presents a new application of BCD for detecting an industrial fault in a wave-soldering machine. The BCD formulation is used to identify the point at which flow rate reductions happen as a result of the pressure drop caused by solder build-up in a pipe. It is assessed on a test bench analogous to a pump system in an industrial wave-soldering machine. A low-cost, non-disruptive monitoring device is used to collect pump vibration data to monitor machine states.

The BCD then successfully generates the likelihood of a change point occurring when the flow rate is reduced. The location of the change point depends upon the features used in the analysis, but it is generally detected either at the start of or during the transitional period leading to significant flow rate reduction. This information can then be used to extract and identify features that indicate the transition towards the faulty behaviour, as well as to provide an indication of the length of machine up-time that leads to faults. These features can help machine operators better identify machine states and inform maintenance procedures before the flow rate, and thus production, is significantly affected.

Following this, an online BCD algorithm adapted from existing literature and applied to the industrial use case. A test bench analogue is used to simulate pipe blockage due to solder build-up in a wave-soldering machine. The pipe state is indirectly monitored using a cheap, non-disruptive device that collects vibration data from the pump feeding the system. The collected data has two features extracted from it, the maximum pump vibrational amplitude and peak frequency as the valve is closed. The OBCD algorithm then generates the likelihood of a change point having occurred in the flow rate as a result of the pipe blockage. This information is processed online and can be used to identify the current machine state (differentiating between a faulty and non-faulty state) and informs machine operators of the need for maintenance.

Both the offline and online BCD algorithms allow for the real time response to faulty machine behaviour before the flow rate is too greatly affected by the valve closure. Thus, they can reduce the number of unsoldered products released by the machine. The offline BCD in particular can perform a successful analysis even with a small data set, and both require minimal system knowledge and little-to-no additional data sets for training or calibration. This makes them particularly suitable for the intended generic application across a machine fleet.

Chapter 6

Industrial Application and Validation

6.1 Introduction

This chapter presents the implementation and validation of the previously presented contributions using Siemens data. The Mixed Integer Linear Programming (MILP) scheduling algorithm is tested with machine processing times representative of those found in the factory. It is compared against a Priority Rules Based Method (PRBM) representative of the factory scheduler. The offline and online Bayesian Change point Detection (BCD and OBCD) algorithms are tested on a limited data set collected from the Siemens Congleton site and also on a number of artificial data sets intended to replicate a variety of machine behaviours. Initial steps towards data collection from Siemens factory Congleton was started but could not be completed as a result of COVID-related delays. However, the first steps towards integration with the existing Industrial Internet of Things (IIoT) network were taken. Finally, the combination of both systems integrated in a factory environment is discussed. The scheduling algorithm can provide an overall global schedule as the fault detection method monitors the machines so that any anomalies or faults can

be responded to quickly. The production schedule can then be updated locally to reflect these issues and recover from them quickly and effectively.

6.2 Modular Production Line Scheduling Algorithm Validation

6.2.1 Methodology

The scheduling algorithm is tested on anonymised machine data and production timings from the Siemens Congleton production line. It includes both the product processing times and the sequence dependant setup times (referred to as "changeover times" in the factory). The factory setup in 4.2 is used as it was agreed with Siemens that it was a suitable representation of the existing production line for the purposes of the validation, assuming that the inspection times are accounted for in the production times. Two product variants, with effectively identical production times, that are processed on the same line have been selected. The same cost function is used to optimise the schedule, as is the same PRBM for comparison.

As in Chapter 4, a set of 100 products is generated, with random product types and due dates. These are scheduled for processing on the full production line and the results are compared to the schedule generated by the PRBM. It is assumed that there are no unplanned machine failures. Product failures are also ignored: in the factory environment, any products that require touch-ups or repairs are processed on a separate production line instead of being rescheduled or reprocessed on the same line.

Additionally, production schedules for separate single, serial and parallel machines setups are generated using the individual algorithms. These will use the same production and setup times used in the full factory simulation. New due dates will be generated, as those used for the full factory simulation are unsuitable. This is intended to demonstrate the capability of the individual algorithms to generate local schedules, as well as the flexibility of the approach to conform to any factory setup. In this way the global schedule generated

by the full factory simulation can provide an overall optimal or near-optimal schedule, and any local disturbances (such as the aforementioned unplanned machine downtime) can be recovered from quickly and effectively.

6.2.2 Results and Discussion

The MILP scheduling algorithm was run on both the production line setup and the individual machine setups using the times generated using the anonymised factory times. This is intended to show that the scheduler is flexible and a generic solution to scheduling in general. A global factory schedule can be generated, or local schedules suited to specific machine setups.

The results of the individual machine setups is shown in Table 6.1. The results follow the patterns shown in Chapter 4. The single machine setup reduces the cost function by 3%, smaller than the improvements shown earlier as a result of the relatively smaller setup times. The longest delay and total processing times are improved by 23% and 3% respectively. However, as a trade off, the average delay doubles and the number of delays increases from 10% to 18% of the total number of products. The serial machine setup shows no improvement between the PRBM and MILP as a result of the time constraints applied to solver and the high complexity of the problem. With additional time or a smaller group of products, the MILP would be able to provide an better schedule. The parallel machine setup shows a significant improvement in the total processing time - a 16% reduction - and a halving of the average delay, but the longest delay and number of delays more than double. The cost function is still improves by 3%. Adjustments can be made to the cost function as required to shift the focus of the MILP, for example to reduce the number of delays instead of the total processing time.

The results of the production line simulation are shown in Table 6.2 and correspond with the findings in Chapter 4: in all metrics bar longest delay, the

results of the MILP are an improvement over the PRBM. The average delay and number of delays improve by approximately 37% each. The total processing time decreases by 21%, and the cost function shows a reduction of 15%. The longest delay increases by 28%.

	Single		Serial		Parallel	
	PRBM	MILP	PRBM	MILP	PRBM	MILP
Average Delay	0.06	0.12	1.41	1.41	0.12	0.06
Longest Delay	1.7	1.3	23	23	5	12
Number of Delays	10	18	19	19	6	15
Total Processing Time	105.8	102.5	575	575	246	206.5
Cost Function	55.06	53.2	313.82	313.82	130.55	126.15

Table 6.1: Validation Simulation Results: Single, serial and parallel machine setups

	Production Line Simulation Results	
	PRBM	MILP
Average Delay	4.81	3.02
Longest Delay	37.3	47.8
Number of Delays	24	15
Total Processing Time	616.3	485.8
Cost Function	378.8	322.03

Table 6.2: Validation Simulation Results: Production line setup

The potential improvements in the overall production schedule are significant, but might be exaggerated by the particular setup used. Exact processing and setup times will affect the results, as will any additional rules used by Siemens' PRBM. Despite this, the benefit of using the MILP formulation will still offer an improvement to the existing system, since it searches for the optimal schedule and cannot perform worse than the PRBM. The additional flexibility afforded by the customisable cost function and modular MILP design is a significant improvement over the static PRBM.

6.3 BCD Fault Detection and Health Monitoring Validation

6.3.1 Background and Methodology

The offline and online BCD algorithms were intended to use data collected from the wave-soldering machine use case. The kit used to gather data from the test bench has been adapted to function in a plug-and-play manner, with minimal requirements for oversight. As a result of data security concerns, the kit could not be programmed to send the collected data samples offsite using the local internet connection. This meant that it had to stay on site over a period that included several maintenance cycles before being returned for processing. The security concerns centred around the fact that this kit was a non-Siemens development kit and thus potentially insecure. Such a device could have provided an access point for malicious software or agents, compromising the security of the network and possibly risking sensitive Siemens data or opening the network up to attack.

Figure 6.1 is an overview of the basic processes of the wave soldering machine. Incoming boards are loaded on a conveyor belt. Flux is applied, in this case by spray, though foam is also used. Flux is intended to remove oxides from the board, improve heat transmittance and provide a smoother soldering process. The board is then preheated to activate the flux, evaporate volatiles and reduce the thermal shock as the board meets the solder. The boards then travel over two solder waves. The first is a chip wave, which provides a turbulent, fast flowing wave intended to ensure that the liquid solder reaches all of the chip components and joints. This is followed by a λ (Lambda) wave, which has an asymmetrical, laminar and slow flow. It completes the formation of solder fillets and removes excess solder, preventing bridging between components. Finally, soldered boards have to cool at a rate that avoids thermal shock, though

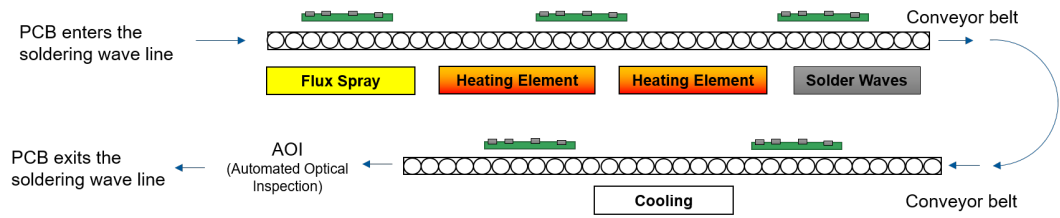


Figure 6.1: *Wave Soldering Line Processes*

the time spend on this is minimised as much as possible. In most cases cooling takes place on a conveyor belt below the flux, preheating and soldering processes. On the use case site, soldered PCBs pass through Automated Optical Inspection (AOI) to identify any fault, before they exit the wave soldering line.

The most common production faults are a result of low solder wave height, as described in Chapter 3. As per [188], these faults can take a variety of forms. The most common of these will be outlined here. The first is insufficient solder topside fillets, in which joints are not sufficiently soldered in place (Figure 6.2a). Similarly, insufficient solder bottom side fillets (Figure 6.2b) describe a lack of solder on the reverse of the board. Finally, solder skips (Figure 6.2c) describe the situation where the component has not been soldered during the soldering process. It is worth noting that a variety of factors can contribute to the occurrence of some of these faults. However, most of these other factors are associated with static machine settings or board design, and addressing them falls outside of the scope of this thesis.

The wave soldering machine has two separate solder pumps connected to the same reservoir. One supports the chip wave and the other supports the λ wave. Figure 6.3 shows the connected accelerometers, placed in 3D printed cases to avoid direct contact with the pumps which operate at temperatures outside of the accelerometers' operating range. Permanent magnets are used to secure them to the pumps [144] instead of the bees-wax used for the test bench as a result of the high temperatures ($>100^{\circ}\text{C}$). The resonant frequency of a magnet mounting is about 7kHz, which is significantly above the expected pump frequencies and as such will not affect the measurements. Accelerometer

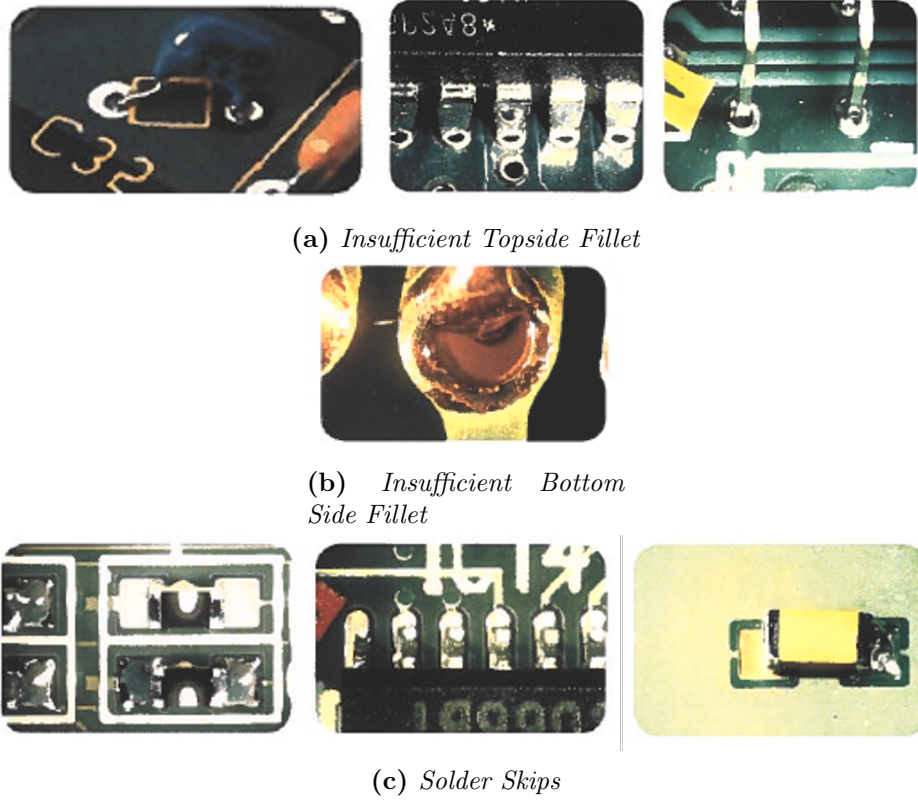
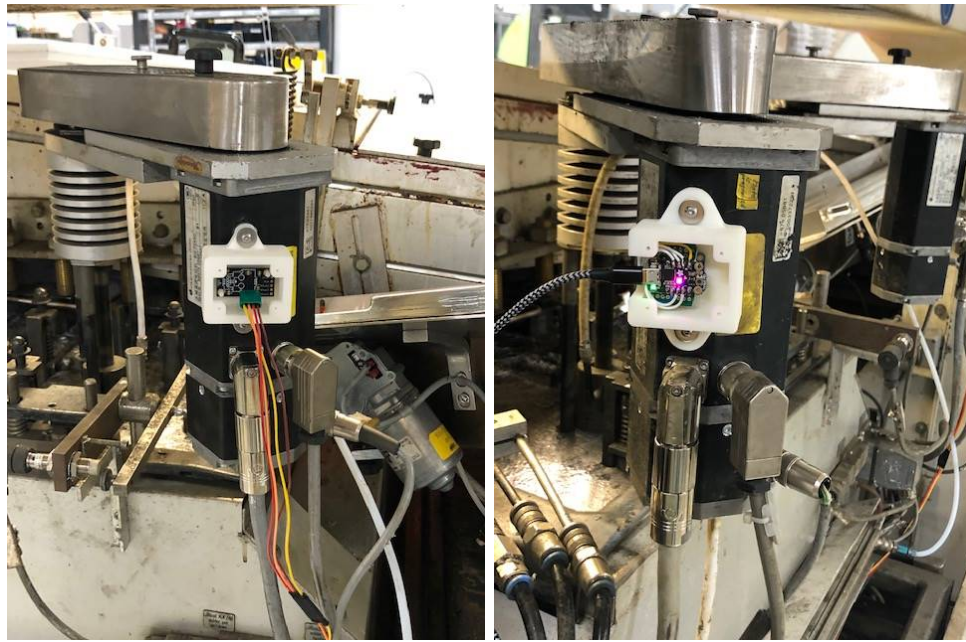


Figure 6.2: *Notable Faults Caused By Wave Soldering Process* [188]

1 (Figure 6.3a) is the one used in the test bench, LSM9DS1. The range is set to $\pm 2g$ to maximise the sensitivity, which is 0.061 mg/LSB. Accelerometer 2 (Figure 6.3b) is an LIS3DH accelerometer set to the same range and sensitivity as pump behaviour is not significantly different across both pumps. The maximum frequency range of both is capped by the capacity of the Raspberry Pi used to collect the data, at approximately 350Hz. Baseline data for the wave soldering machine pumps has a listed baseline setpoint of 1600rpm [189], which can be converted to 26.7Hz. This is well below the Nyquist limit of the sampling frequency [190].

The initial data collection process collected a sample every half hour over the course of a month. Each sample recorded approximately 30 seconds of vibration data made up of 10,000 data points collected at approximately 350Hz, and processed using a Fast Fourier Transform (FFT) to get the corresponding amplitude and frequencies. The maximum amplitude and peak frequencies



(a) Accelerometer 1 on Machine Pump 1 (b) Accelerometer 2 on Machine Pump 2

Figure 6.3: Accelerometers on the Machine Pumps

extracted from these samples is plotted in Figure 6.4. The plot appears to show a number of clear transitions between different states. However, initial analysis suggested that these change points did not correlate with any events logged by Siemens. Further pump analysis also showed that the pump operates at different power levels dependant on the current product type being processed. This changes the wave heights to suit the design of the product's board. As the wave soldering machine processes multiple product types over the course of a half-hour period, the pump power levels will likely have changed multiple times over said period and a lot of information is lost.

To compensate for this, the data collection process was adjusted such that samples were collected every minute, with the features extracted immediately after collection to reduce the amount of data storage required. Unfortunately, as a result of COVID and related delays, it was impossible to complete the required data collection in time for analysis here.

Additionally, the machine's production schedule/maintenance timings were unavailable in the necessary time frame, limiting the potential of the BCD as well as the ability to accurately validate its performance. This information would

have allowed the data to be filtered such that only readings from a single pump power level were present for analysis. Alternatively, the readings could have been normalised with respect to the pump power level, potentially creating a larger, more cohesive data set and reducing the risk of missing features indicative of a fault.

Without this information, a best guess could to be made to filter out the pump behaviour at a particular power level and attempt to locate a change point indicating a fault. However, if the filter band is too narrow, it could remove the fault-indicating feature. If the band is too wide, it could include changes between different power levels, which the BCD algorithm might pick out instead. This limits the BCD to segmenting the data and potentially replicating information available in the production schedule. The exact pump power levels are recorded locally, but have to be accessed and extracted physically from the machine. This is a form of data silo and is likely to be present in a large number of machines around the factory, especially ones with proprietary software, unique data types, or lacking up-to-date IoT connectivity.

For this reason, four artificial data sets have been generated to test both BCD methods. These data sets are intended to mimic pump responses to different conditions. They represent either a filtered pump data set, such that only one power level is represented, or potentially a normalised data set that adjusts each data point proportionally to the pump power level at the time of its observation. As such, the maximum natural pump response is capped at 100, to represent pump operation at 100% power. Random noise with a uniform distribution of between 10 and -10 is added to each of the data sets to mimic the effect of random industrial vibrations. The specific conditions and the values used to generate each data sets will be outlined below.

The first data set, shown in Figure 6.5a, mimics the flow rate drop experienced by the test bench: the pump response begins at 100 and experiences a slight linear drop. This is followed by a transitional period, and a larger linear drop

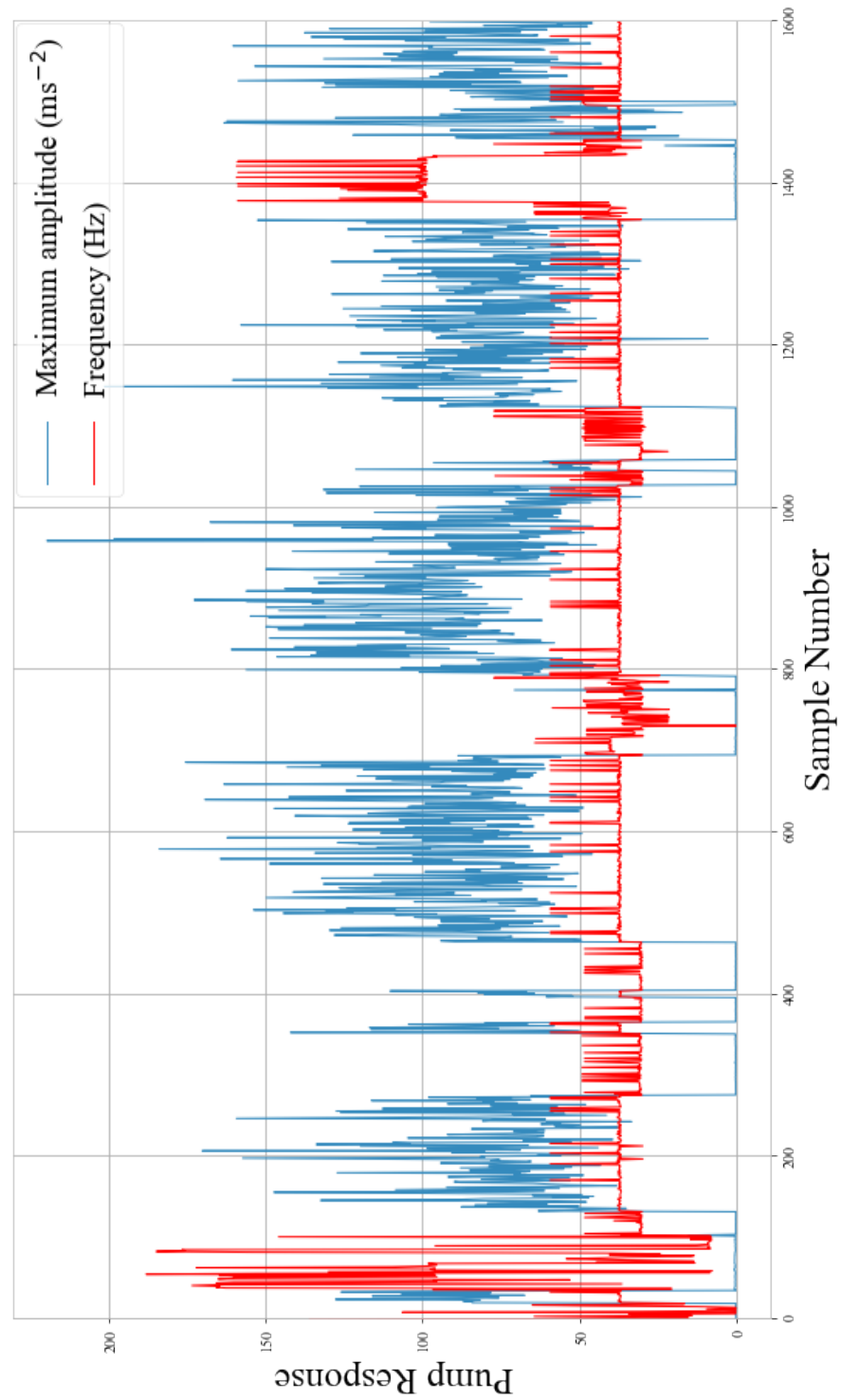


Figure 6.4: Features extracted from half-hourly samples collected from Machine Pump 1

to 0. The initial linear drop has a gradient of -0.1 and the second linear drop has a gradient of -3.33. The transition occurs between the times 65-75.

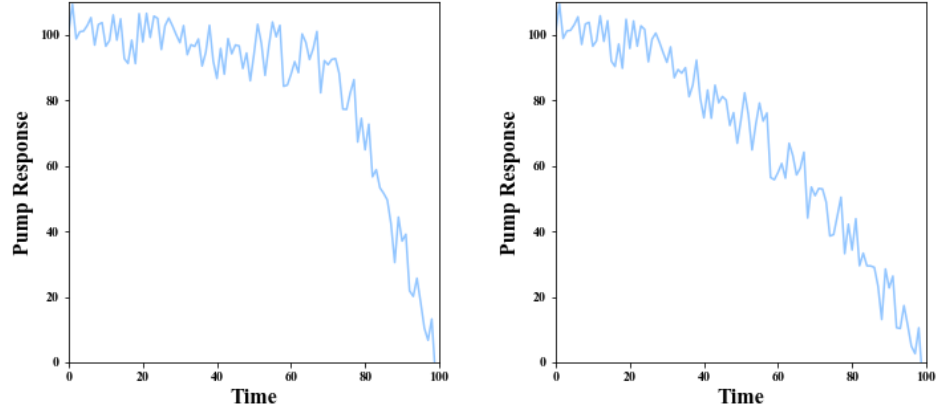
The second data set, Figure 6.5b, assumes that the pump response is proportional to the change in internal clearance as solder dross build up over time. It takes the form of a stretched quadratic curve, starting at (0,100) and ending at (100,0).

The third data set, Figure 6.5c, assumes the pump response changes at a point in time as a result of a sudden change in the system. This could be a result of a number of events, such as leakage, bearing failure or seal failure. The pump response drops from 60 to 50 at time 70.

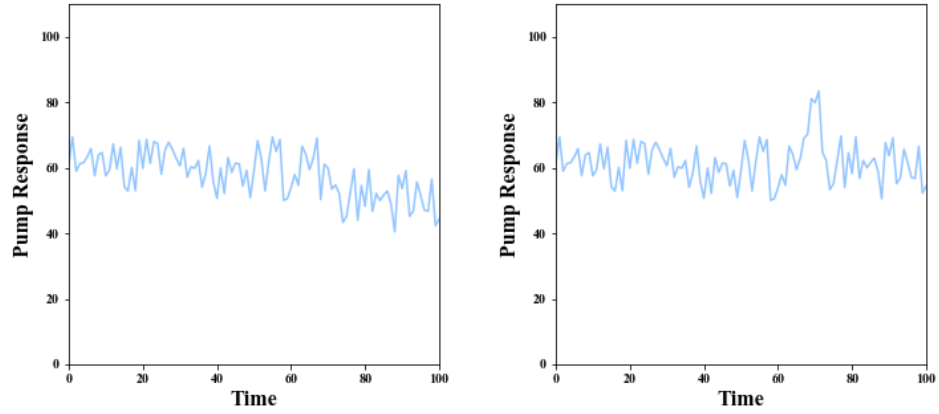
The final data set, Figure 6.5d, represents a temporary, anomalous change in output. This could be a result of a temporary blockage, nearby disturbance or other short-term event. In this case, the pump response changes from 60 to 80 and back over the times 68-72.

The goal of the offline BCD algorithm is to retroactively analyse data generated by a machine and segment it such that the change from healthy to faulty behaviour can be identified. This allows features indicating upcoming or current faults to be extracted so that they can be compared to online data and used to prevent or reduce the effects of faulty behaviour. Additionally, the location of the change point can give users an indication of the expected length of machine uptime that would lead to a fault. The results of the BCD would supplement and be compared to maintenance and repair logs to assess and improve their accuracy. In this case, the results will be compared to the parameters used to generate the data sets.

The goal of the online BCD algorithm is to provide a real-time indication of the current machine state. It segments the behaviours of the pump based on an initial model and provides information about its confidence in the existence a change point, as well as the change point location. The fitted model can also



(a) *Pump Response Proportional to Test Bench Flow Rate* (b) *Pump Response Proportional to Area Reduction as a Result of Solder Build Up*



(c) *Pump Response During Step Change in Output* (d) *Pump Response During Temporary Anomaly*

Figure 6.5: *Artificial Data set Generated to Assess Online and Offline BCD Performance*

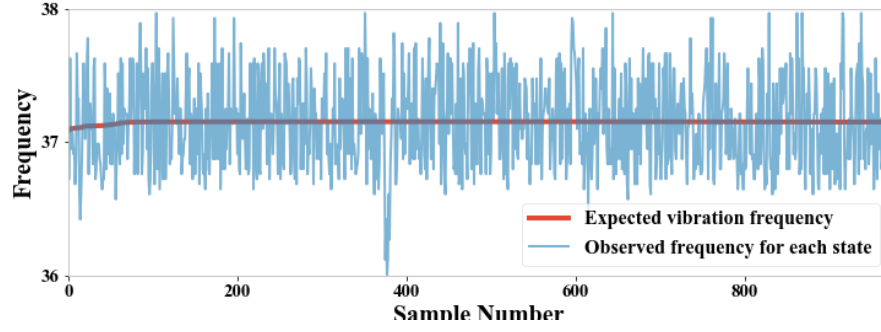
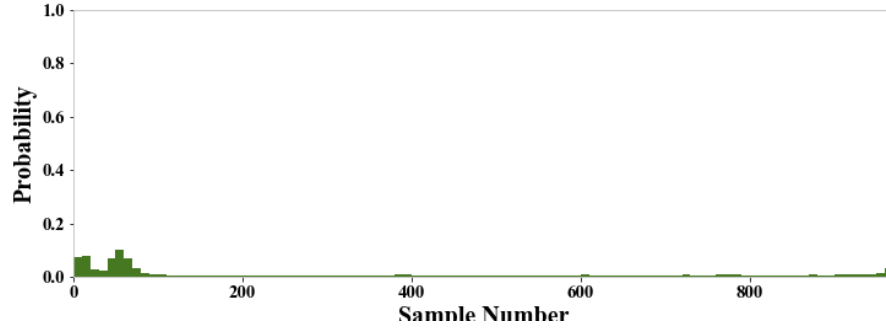
let users know what behaviour to expect from the machine given its current state. A comparison afterwards to the maintenance schedule and repairs logs can be used to provide an indication of its accuracy. As with the offline BCD, these results will be assessed through comparison with the parameters used to generate the data sets.

6.3.2 Results and Discussion

Offline BCD

An initial attempt to analyse the collected Siemens data is presented first. Figure 6.6a shows the frequency data filtered to present a best guess of a single pump power level, and the associated results of an offline BCD analysis. This subset of data was selected by filtering the frequency across a band between 36 and 38 Hz. The band was chosen as it was the most commonly occurring frequency band detected across the samples collected from the factory. The expected frequency varies very little across the sample. Figure 6.6b suggests that the predicted change point occurs very early in the data, but has a low level of certainty, with low values of τ spread across the range of samples. The most likely reasons for this result are the infrequent sampling of the vibrations, and the lack of information about the timing of maintenance procedures. Infrequent sampling can lead to unrecorded changes in pump behaviour. Lack of information about maintenance timings means that the data set can't be split to reflect maintenance taking place, which would reset pump behaviour and add one or more change points to the data set. Another potential factor for the low certainty is the filtering band being too tight and removing the change point-indicative pump behaviour, or too loose and capturing the behaviour of multiple pump power levels. This could potentially be overcome with a significant amount of tuning, although it would be difficult to validate any results without additional information. Additionally, as discussed above, a wider filtering band could catch multiple pump power levels resulting from different product types. As such, the best expected outcome without supplementary data for this use case is that the pump behaviour could be segmented by power level.

The results of the offline analysis on data set 1 are presented in Figure 6.7a. The BCD algorithm located a change point with high confidence within the

(a) *Observed and Expected Frequencies*(b) *Predicted Change Point Location and Likelihood***Figure 6.6:** *Offline BCD Analysis on Filtered Machine Pump 1 Data*

defined transition zone, with results similar to those presented in Chapter 5. It also correctly inferred the model parameters that were used to generate the data set. The inferred model can then be used as a reference for comparing incoming data to in order to assess the machine health, as well as providing indication of the length of machine up-time during which the behaviour can be assumed to stay healthy.

The results from data set 2 are presented in Figure 6.7b. Initial analysis bisected the data set with a low-confidence change point spread over a wide margin over the middle of the data set. However, at this point the pump response has changed significantly and the flow rate would likely also be significantly impacted. Assuming that it is necessary to detect a change point before this takes place, the BCD algorithm can be tuned to split the data set such that the initial pump behaviour is defined by a shallower response over time. The results shown in Figure 6.7b are a result of this tuning, and locate a change point with medium confidence early in the data set, before a significant

change in the pump response. This highlights a point of interest about the use of BCD analysis and the effect the choice of priors on the results of the analysis. Adjustments to the priors can significantly affect results and as such should be done based on the requirements of the analysis and problem being solved. However, this is also an important benefit of this approach as well, as it allows the analysis to be improved based on early results and existing knowledge about the problem.

The results from data set 3 are presented in Figure 6.7c. In this case, the BCD algorithm inferred the location of the change point with high confidence and high accuracy. The pump behaviour both before and after the change point also closely matched the expected values. As with data set 1, this provides a reference for the expected machine behaviour, as well as the behaviour in the case of the fault being displayed. This knowledge could then be applied to monitor the machine in real time with comparisons to the expected and actual behaviour, and assist troubleshooting when faults take place.

The first set of results for data set 4 are presented in Figure 6.7d. This initial analysis did not detect a change point with any appreciable degree of confidence, but did infer a model close to the parameters used to generate the pump behaviour either side of the change point. This result has value as a means of filtering out short-term anomalous signals, if existing knowledge about the system believes this to be appropriate. Otherwise, there are two other means of analysing the data set if it is important to locate this anomaly and establish its effect on pump behaviour. The first of these is to split the data set such that it ends during the anomalous behaviour. This is presented in Figure 6.8a and demonstrates that, even with a very limited sample size, BCD can confidently and accurately locate the change point representing the start of the anomalous behaviour. The second method is to adjust the BCD to search for two change points, as shown in Figure 6.8b. This version of the algorithm accurately and confidently locates the both the start and end of the anomaly. The approach

used in this type of case depends upon the requirements of the scenario.

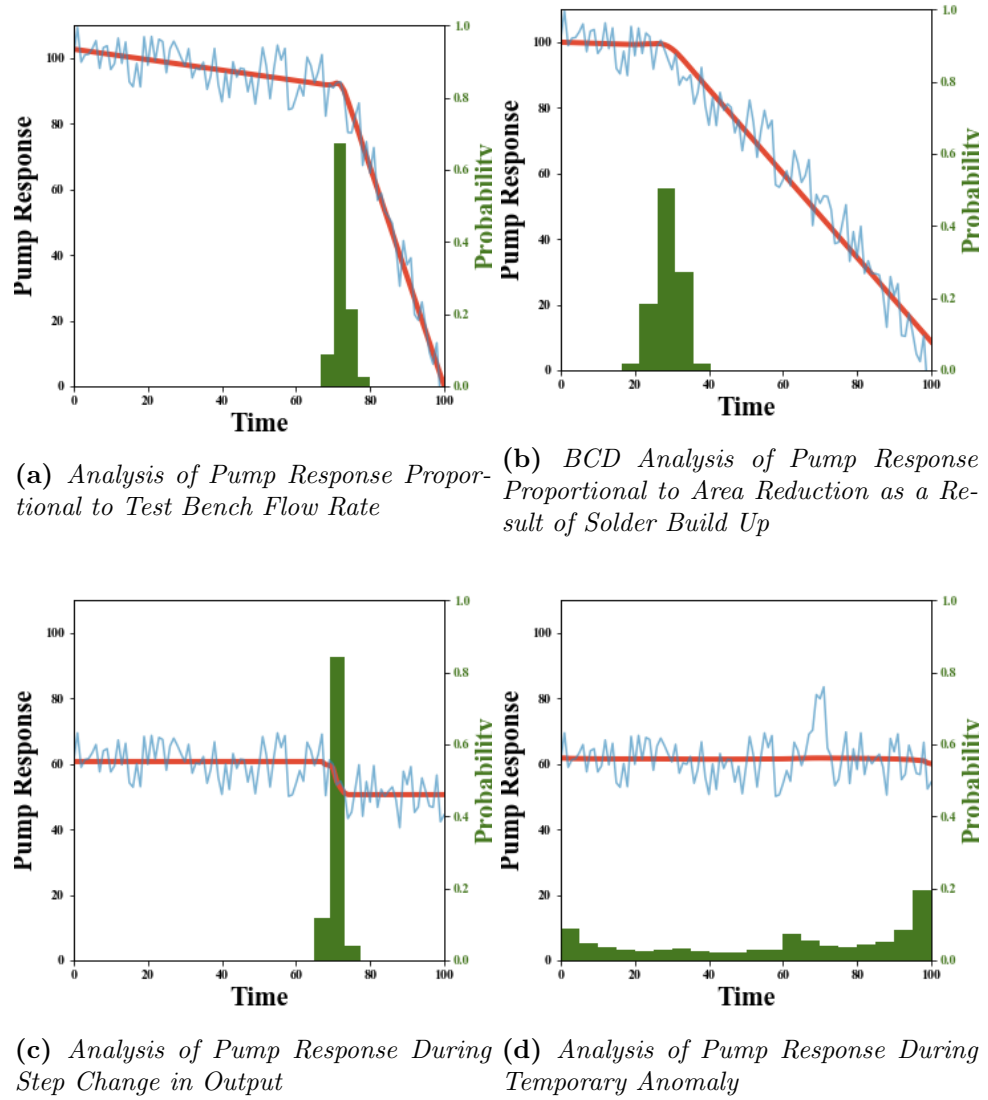
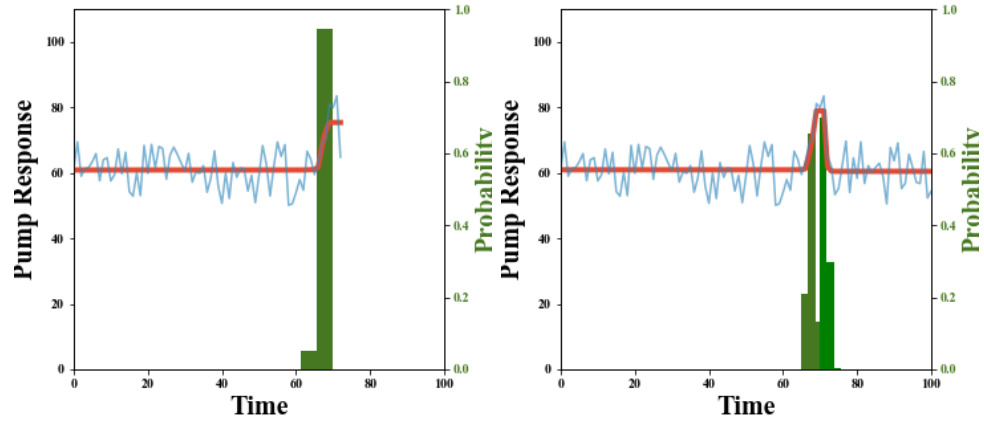


Figure 6.7: Offline BCD Analysis of Artificial Data set. Blue: Observed data. Red: Inferred behaviour. Green: Location and likelihood of a change point.

Overall, the offline BCD performed well and successfully identified change points in the artificial data sets. It was able to infer models to fit the pump behaviour that reflected the parameters used to generate the data sets. With a more complex behaviour like the quadratic curve in data set 2, the inferred model fails to accurately capture the behaviour but does provide a useful approximation of it alongside early detection of the flow rate drop. The inferred model parameters can be extracted and used as reference points for comparison in future FD-HM applications as well, creating a library of machine behaviours



(a) Analysis of Split Temporary Anomaly Data set (b) Analysis With Two Change points of Temporary Anomaly Data set

Figure 6.8: Supplemental Offline BCD Analysis of Temporary Anomaly Data set. Blue: Observed data. Red: Inferred behaviour. Green: Location and likelihood of a change point.

representative of different machine states. This can facilitate new monitoring techniques in the future.

Data sets 2 and 4 highlighted the most important potential weakness of the BCD analysis: incorrect priors and assumptions about the model. These can produce misleading results (as per the familiar idiom “rubbish in, rubbish out”), but analysis of the results of the Markov Chain Monte Carlo (MCMC) processes and the generated posterior distributions can indicate that the formulation is incorrect and needs further development. It is for this reason that Bayesian data analysis problems often include an evaluation and analysis stage to ensure that any conclusions drawn from the results are meaningful. As such, while it is true that an incorrect formulation will undermine the accuracy of any insights gleaned from the initial BCD analysis, the accepted best practice of incorporating a results evaluation and analysis stage as part of a cycle of continuous improvement for the model will help to counteract this.

Online BCD

Analysing the Siemens data using the online BCD algorithm yields similar results to those generated by the offline algorithm. Figure 6.9 shows that the inferred model doesn't change over time and no change point is detected at all. It is worth noting that as the first of the samples are being processed, and the initial behaviour models are being inferred, some change points are detected with low-medium confidence. A user might be notified of these as they happen, but they generally disappeared as the behaviour models updated. Additionally, online BCD requires the use of a shifting margin to keep the sample size, and thus processing time, low enough to run in real time. Without access to processing or maintenance information, it is difficult to select an appropriate size for the margin.

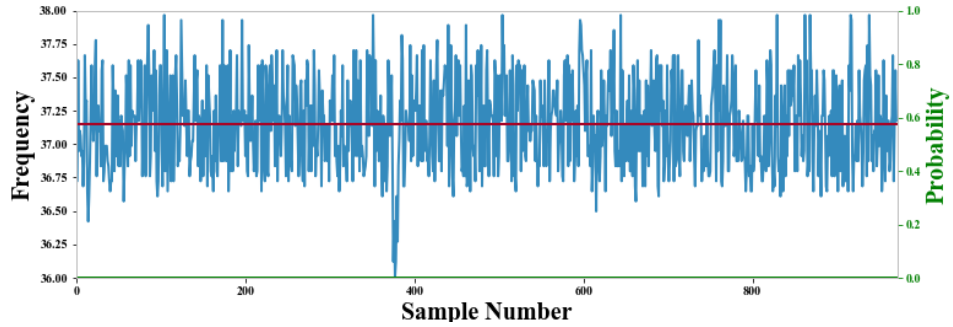


Figure 6.9: *Online BCD Analysis on Filtered Machine Pump 1 Data. Blue: Observed data. Red: Inferred behaviour. Green: Location and likelihood of a change point.*

The results of the online analysis on data set 1 are shown in Figure 6.10a. The inferred behaviour closely follows the observed data and the change point is identified at the right point, with the highest confidence near the end of the transition period. While this doesn't provide preemptive warning about the incoming change in flow rate behaviour, it will allow users to react more quickly to the change and also assist in troubleshooting the fault should there be any uncertainty about the cause.

The results from data set 2 are shown in Figure 6.10b. The location of the

change point is spread across a range of times around the middle of the data set, without a high level of confidence. As the behaviour model is inferred based on incoming data, it is more difficult to tune the online BCD to achieve an earlier response like the one presented in the offline analysis of this data set. However, this could be overcome by responding to the changes earlier, when fewer data points have been collected.

The results from data set 3 are shown in Figure 6.7c. The inferred behaviours either side of the change point match the parameters used to generate the data, but the location of the change point itself is less certain and spread over a small range of times. As with data set 1, there is no preemptive warning, but occurrence of the change point will notify users and new inferred behaviour can be compared to the existing behaviour to assist troubleshooting.

The results of data set 4 are presented in Figure 6.10d. As with the offline BDC, the end result of the analysis of this data set does not effectively detect a change point at the end of the time, with an almost constant inferred behaviour model and broad, low confidence predictions about the change point location. However, during the analysis, as the time advanced past the start of the anomaly, a change point indicating the start of anomalous behaviour was detected with high confidence, though the location was placed early as a result of the noise. This is shown in Figure 6.11. In an industrial environment, a user could be notified of this and would have a chance to assess the situation before the change point was filtered out.

To summarise, the online BCD algorithm performed similarly to the offline algorithm, successfully analysing the data sets and locating change points, as well as inferring models for the pump behaviour. Its performance for data set 2 was worse, although this issue could potentially be overcome with additional knowledge about the system. The analysis of data set 4 is better, as it can notify a user that a temporary anomaly has occurred but in cases where performance returns to normal afterward it can also filter it out. As well as the

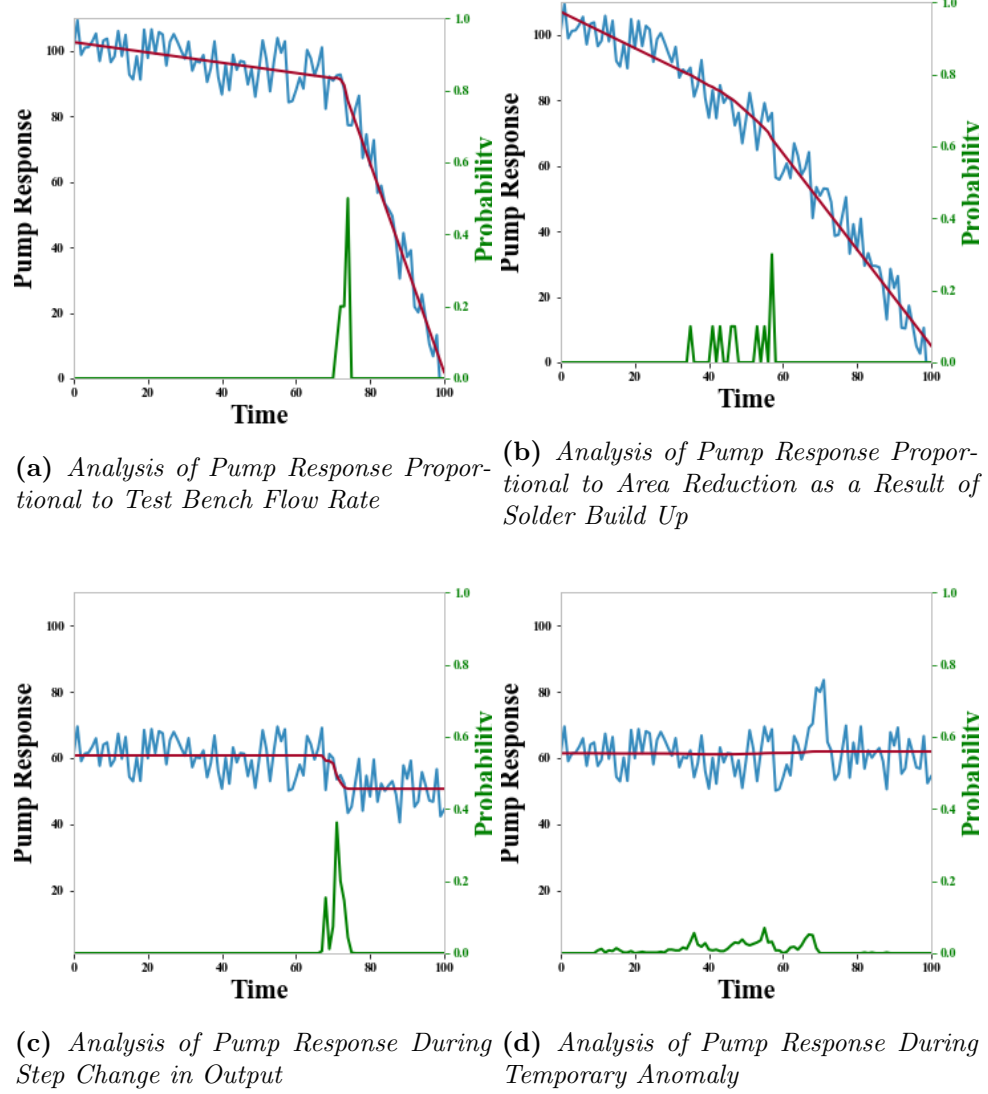


Figure 6.10: Online BCD Analysis of Artificial Data set. Blue: Observed data. Red: Inferred behaviour. Green: Location and likelihood of a change point.

notification of the location of the change point, the inferred behaviour model generated in real time can be compared to the models generated by the offline analysis in order to check what behaviour the machine in question is currently displaying.

Similarly to the drawbacks of the offline BCD algorithm, the effectiveness of the online BCD algorithm relies heavily on the assumptions made about the data models. Additionally, to ensure timely responses to incoming data, the sample size has to be kept relatively small. This can be done using a shifting margin or window; or by beginning a new data set after a change point has been

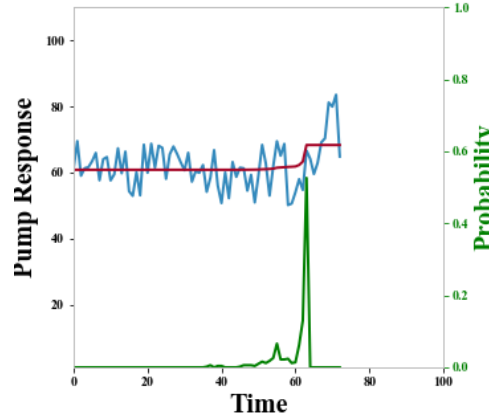


Figure 6.11: *Online BCD Analysis of Pump Response During Temporary Anomaly: live warning about change point taking place as a result of the anomaly.*

detected and confirmed. These approaches rely on supplementary information about the system to inform either the window size or to confirm the change points and begin a new data set.

6.4 Integration with IIoT and MindSphere

A secondary goal of this project was to integrate the data collection and processing methods with Siemens IIoT and cloud platform, MindSphere [47]. The aim was to take the “wrap and re-use” approach [12] to digitisation, extracting additional value from existing machines with a low cost, generic solution. Sensors attached to a machine would collect data from it and send that data to a local edge device, which could then upload the data to the cloud platform. Once the data was uploaded and processed, it could be disseminated to relevant users quickly and easily, and remain available to access for interested parties. Product Lifecycle Management (PLM) software such as Siemens Teamcenter [191] could be used to store this data, supplementing any existing information, or the data could be used to form a historical database, allowing for other forms of analysis to be performed in the future.

Figure 6.12 presents a topology diagram of the information flow across the

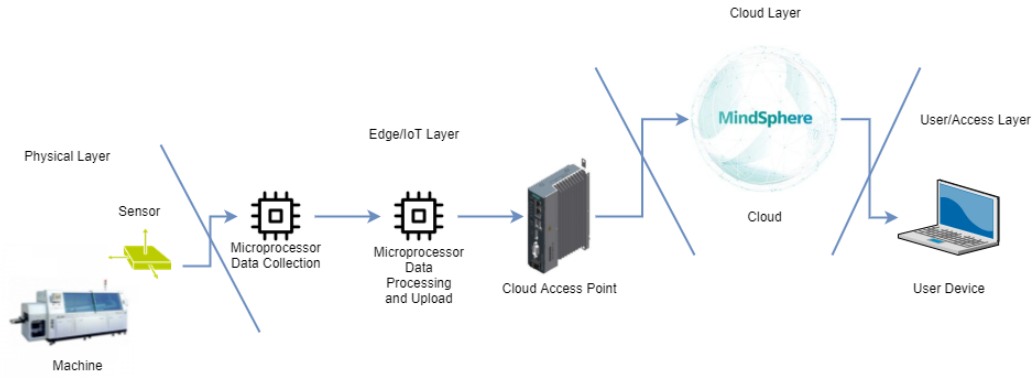


Figure 6.12: *Topology Diagram: Information flow from machine to cloud*

information layers from the physical layer to the cloud layer. On the physical layer, a machine generates the data, such as motor or pump vibrations, or current and voltage draw. A suitable sensor or sensor suite collects the data, connected to a microprocessor or microcomputer at the Edge/IIoT layer. From here, some initial processing can be done to the data, to reduce its overall volume, or it can directly be sent to a local cloud access point and uploaded to the cloud platform. Once on the cloud platform the data can be analysed, stored and disseminated either as active notifications to relevant parties or as a passive source of information about the state of the machine.

As mentioned earlier this chapter, access to production data and schedules to supplement the machine data analysis can significantly improve the quality of the conclusions that can be drawn from the data. Figure 6.13 shows the ideal data flow which combines a production schedule and data from the MES (Manufacturing Execution System) with the data gathered from the machine. This would allow sensor outputs to be matched to particular machine behaviours with confidence, as well as maintenance times or unplanned machine down time. User response could be tailored to a particular machine, its expected behaviour and observed behaviour. This information could also be used to update production schedules in cases of disruptive faults and help production recover faster and more smoothly.

Integration with IIoT networks and data integration in general aims to con-

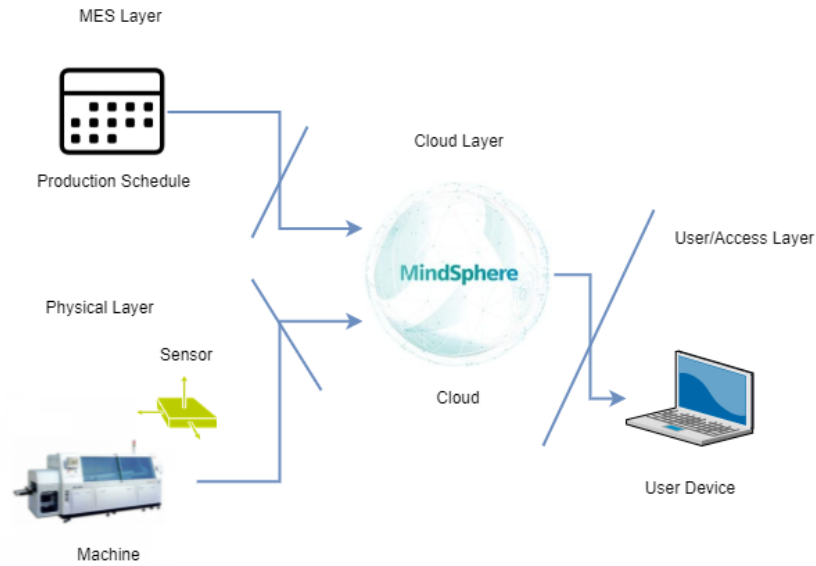


Figure 6.13: *Topology Diagram: Supplemental data collection*

solidate disparate data sources, process the collected information and then make it available through a variety of sources. The data processing includes cleansing and reformatting, as well as an assessment of the data quality [34]. This data can then be made accessible for self service through a variety of devices to suit the needs of individual users. Overall, successful integration can improve collaboration, save time, reduce rework and increase the value of available data.

Conversely, a number of risks have to be addressed when looking to integrate IIoT infrastructures [40]. Chief among these is the security and privacy of the network [55]. IIoT networks are often vulnerable to a number of attacks, including Denial-of-Service (DoS) attacks, physical damage and eavesdropping. Security mechanisms can be implemented to help avoid this, such as: protocol and network security that offers an end-to-end secure communication mechanism; identity management for authentication and authorisation; and fault tolerance for preventing and detecting attacks. Latency and interoperability are also key issues to tackle to ensure full value can be extracted from data. Reducing latency is vital in time-sensitive processes that require quick reactions to inputs, such as location awareness and navigation or real-time user responses. Interoperability is being addressed through the creation of

new standards to enable the creation of smart scenarios despite heterogeneous devices.

In order to integrate the current data gathering setup and fault detection algorithms with the factory IIoT and MindSphere network, the following steps would have to be taken. The Raspberry Pi 3B currently in use is not industry standard and would need to be upgraded. Options for this include Siemens' SIMATIC IOT2050 [192] and the Revolution Pi [193], which are both certified for use in an industrial environment, coming in DIN-rail housing for easy installation. In some cases, a secondary device, such as the MindConnect Nano [194] will be required to collect the data output from the primary data-gathering device to process and upload it at a local cloud access point. This might be necessary in cases where edge apps prevent access to local physical inputs such as USB ports. However, this solution does have the benefit of allowing the device responsible for data collection to focus exclusively on that. A potential drawback is the increased risk to security through the addition of extra IIoT nodes; however, this system is isolated and difficult to access physically without permission. Communication can be done through OPC-UA or MQTT protocols, depending on local requirements: in the case of this project, OPC-UA is almost ubiquitously supported [194]. An important restriction to note is that the MindConnect Nano can currently upload a maximum of 250 data points per second [194].

Once uploaded to MindSphere, the data can be stored and analysed through a custom app that performs the online and offline BCD analyses. These results can be stored online and made accessible to users as and when needed. They can take the form of current, up-to-date graphs, notifications about changes in machine state, and historical data for further analysis and comparison. At this point, if a production schedule can be accessed, expected machine states can be extrapolated and used to supplement the BCD analysis as described earlier this chapter.

6.4.1 Combined scheduling and FD-HM

Full integration of the data gathering setup and BCD algorithms with the local IIoT and cloud network could provide the opportunity for a combined scheduling/Fault Detection and Health Monitoring (FD-HM) system. Improved knowledge about the current and upcoming state of machines can significantly improve a production schedule, reducing the chances and effect of unplanned downtime and ensuring maintenance could be performed as and when needed.

This implementation would take the form outlined in Figure 6.14. A global schedule would be generated, taking into account information about machine health in the form of upcoming maintenance requirements. It would then be broken down into a series of local schedules for individual machines or machine groups, which would follow the schedule to complete the necessary production. The FD-HM system would combine the expected machine behaviour with the analysis of the observed machine data and results of product tests. Any unexpected faults would trigger a notification for users and an update to the local schedule for speedy recovery. The global schedule could then be adjusted to compensate in the longer term.

The benefits of a more responsive and informed schedule include better overall production times as well as reduced time-loss to troubleshooting and reacting to unexpected events. Real-time knowledge about machine behaviour can improve response times to faults and aid identification. Notably, this form of data integration across enterprise levels is the intended vision for the MindSphere platform [48]. However, there are currently a number of potential limitations that would need to be addressed. The local data silos created at the machine level mean that the expected machine behaviour has to be extrapolated. If this is not done accurately, it will have knock-on effects on the accuracy FD-HM data analyses. Additionally, data extraction from particular machines might be impossible, or require specific and expensive tools and sensors as a result

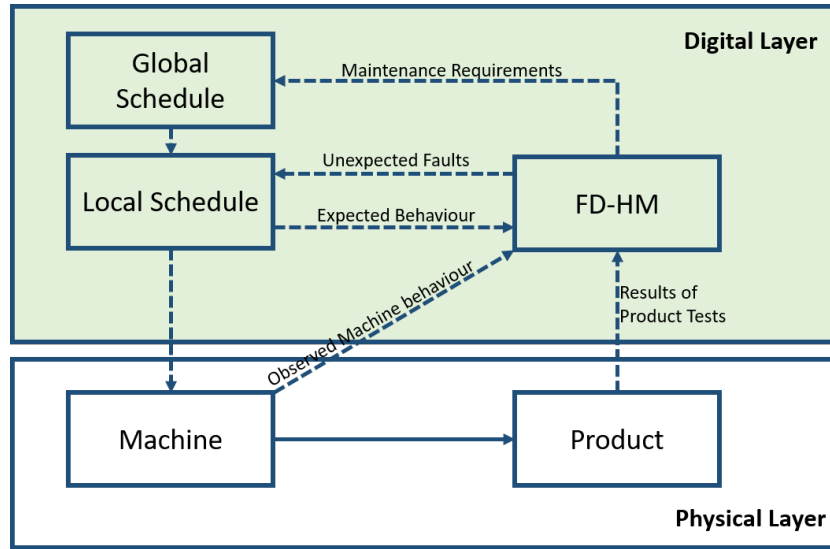


Figure 6.14: *Information Sharing Between Scheduling and FD-HM. Dashed lines represent movement of information.*

of harsh working environments, limited installation space or erratic behaviour. Finally, the volume of data collected and the associated latency are issues with situation-dependant importance and the requirements for both need to be assessed to ensure the integration methods are suitable.

6.5 Concluding Remarks

This chapter focuses on the validation of the contributions developed in this thesis and their integration with an existing factory environment. The MILP scheduling algorithm generates a schedule using anonymised factory data, which is then compared to one generated using a PRBM. The modularity of the MILP is also assessed as each individual setup type is also assessed. Both the factory setup and the individual approaches perform better than the PRBM, aside from the serial machine setup, which was unable to create a better schedule in the time provided.

The offline and online BCD algorithms were tested on artificial data sets representing different potential faults. The results varied but both online and offline analyses could effectively and with high confidence locate the change point in

most of the data sets and infer the parameters that generated the pump responses. The offline analysis was found to filter out temporary anomalies in its current state, but could be adjusted to search for additional change points as required. The online analysis identified the start of the temporary anomaly as it occurred, but could also filter it out if it wasn't addressed or deemed important. Initial attempts at analysing data collected from the wave soldering machine were made and highlighted the difficulties faced by BCD when knowledge about the system was limited.

The initial steps towards factory integration of these methods were presented. The Raspberry Pi currently in use can be upgraded to an industry standard microcomputer, such as the Revolution Pi or Siemens MindConnect Nano. The collected data can be transmitted wirelessly using communication protocols such as OPC-UA or MQTT. This data can then be collected and analysed on the cloud platform MindSphere and the results disseminated rapidly to relevant users.

An approach combining the presented scheduling and FD-HM algorithms was discussed. A global factory/production line schedule can be prepared in advance based on the current factory state. Local updates can then be made based the conditions at particular stages of the production line to improve recovery in cases of unexpected machine downtime. Additionally, knowledge about machine health will feed into maintenance decisions, which will inform the global schedule to ensure full utilisation of available resources.

Chapter 7

Conclusions and Future Work

7.1 Review of Contributions

The focus of this thesis is the non-disruptive leveraging of digital technologies to improve production in an existing factory site. The presented contributions are intended to be generic and applicable to a wide variety of industrial environments. The contributions were then validated using a combination of Siemens data and artificial data sets mimicking situations that industrial equipment might face. Finally, industrial implementation is discussed alongside the integration of both the scheduling algorithm and the fault detection method.

7.1.1 Modular Production Line Scheduling Algorithm

Chapter 4 presents a scheduling algorithm that uses Mixed Integer Linear Programming (MILP) to optimise a production schedule with sequence dependant setup times. Variants are presented that can solve problems involving single, serial and parallel machine setups. The MILP algorithm is tested against a Priority Rules Based Method (PRBM) scheduler analogous to an existing factory scheduler. Both a complete production line and individual machine

setups are considered. For the individual machine setups, sets of 10, 50 and 100 products of two different types are used to generate a schedule. For the complete production line, a set of 100 products of two different types is used. The MILP algorithm is shown to be capable of creating an equal or more efficient schedule with improvements defined by the chosen cost function, which in this case minimises the sum of the maximum delay and average completion time. The cost function can easily be altered to prioritise improvements in other metrics depending on the requirements of the user.

Single Machine Formulation

The single machine formulation is designed to solve the scheduling problem for a single machine processing products with sequence dependant setup times. This formulation forms the basis for the serial and parallel versions. It is tutorialised such that readers will be able to more readily understand and adapt it for specific scenarios. Across the different sizes of sets of products, this formulation reduces the cost function by approximately 20% when compared to the results of the PRBM.

Serial Machine Formulation

The serial machine formulation adds a precedence constraint to ensure that a product cannot be processed on a machine before the prerequisite processes have been completed on the previous machines. This ensures that products are processed in the correct order along the set of serial machines. Additional constraints are also used to calculate the completion times of the first product on all of the machines, all of the products on the first machine, and finally all of the products on machines 2 and above. For the sets of 10 and 50 products, the serial formulation reduces the cost function 15% and 5% respectively. For the set of 100 products, the increased complexity of this formulation meant that the solver could not find a better solution than that of the PRBM in the

time provided. As a result, once the allotted time elapsed, the solver presented a schedule whose performance was equal to that of the PRBM. Increasingly complex or large product sets will produce the same results unless solving time or processing power is increased.

Parallel Machine Formulation

The parallel machine formulation uses a constraint that ensure that products can not be completed before their associated release dates but can be completed on any of the available machines. Additional constraints calculate the completion time of a product based on machine availability, the completion time of the previous product and whether any setup time is required. The highest generated completion time for that product is the one that takes precedence. As with the serial formulation, the additional complexity of the parallel formulation meant that it could not provide a better performance than that of the PRBM within the allotted time. The cost function for the sets of 10 and 50 products is reduced by 27% and 24% respectively.

Production Line Setup

To assess the performance of the scheduling algorithms used in combination, they were assessed on a simplified representation of a production line, consisting of a single machine, followed by 3 machines in series, and finally 3 machines in parallel. To represent the batch transport mechanism used in the factory, the full set of products is processed on the single machine and released as batches of 10 products to the serial machines, with similar batch transport taking place between the serial and parallel machines. In this case, the cost function is reduced by 2%. However, the longest delay shows an increase of 1%. This is a byproduct of the particular cost function in use and its interaction with the single machine schedule. Based on the results of the individual machine groups, this problem could be overcome by assigning due dates for

products at each stage of processing rather instead of a single overall due date. Additionally, the use of a different cost function could have significant impact on the results.

7.1.2 Bayesian change point Detection for Fault Detection and Health Monitoring

Chapter 5 uses a test bench to simulate a wave soldering machine in the Siemens Congleton site. Pump vibrations are recorded as a pipe blockage induced fault is simulated using a gate valve. As the valve is closed, the flow rate is also monitored for comparison and exhibits the following behaviour: a small, linear drop in flow rate until the valve is approximately 70% closed, followed by a transition period, and an almost linear drop to no flow from approximately 80% closed onward. Features are extracted from the pump vibrations using a Fast Fourier Transform (FFT). These are the maximum amplitude and peak frequency of each sample. The Fault Detection and Health Monitoring (FD-HM) methods used here offer two significant advantages over those present in the literature. The first of these is the lack of a need for labelled training data or mathematical behaviour model, as Bayesian Change point Detection (BCD) is data agnostic and infers the expected behaviour using a combination of the observed behaviour and assigned probability distributions. The second is that the indirect monitoring method will function regardless of the location of the blockage in relation to the sensor, a factor that significantly affects the effectiveness of direct pipeline vibration monitoring.

Offline BCD

Offline BCD is used on the features extracted from the pump vibrations to detect the transition from a nominal state to a faulty one. This is done retroactively and successfully identifies a change in the rate of flow rate reduction as

the gate valve closes. The analysis of the extracted amplitude suggests that the change point takes place at the point where the valve is approximately 75% closed, which corresponds with the transition period from low to high flow rate drop. The analysis of the extracted frequency suggests that the change point takes place later, after the transition period. This is partly due to the use of a small number of samples, which does affect the precision of the analysis. However, it also highlights an additional benefit of BCD in that the analysis can still be performed even with a limited number of data points. That said, it is important to keep in mind the effects of the small sample sizes when drawing conclusions from analyses.

As this analysis is retroactive, the key use of the results would be as a means of isolating pump responses indicative of healthy and fault machine behaviour. The BCD algorithm effectively segments the data about the change point into these states, and outputs the features that define them. With this information, the vibration responses of future pump cycles can be assessed to detect said features and inform operators of the current state of machine health and the need to perform maintenance procedures. Additionally, the point at which the change point occurs can also be interpreted as the length of machine up time taken to reach the faulty state. This assumes that the deterioration of the machine is proportional to the length of machine up time, which is the case for the wave soldering machine use case.

Online BCD

An extended test bench data set is then assessed by the the Online BCD (OBCD) algorithm. The OBCD algorithm collects incoming data points from the current machine cycle and compares them to the data already collected this cycle. It infers a model of the expected values and provides an estimate of the likelihood that the incoming data indicates a change point. This can supplement the results of the offline BCD algorithm by providing a real-time

indication of the machine state to machine operators. As with the offline analysis, analysis of the amplitude detects a change point earlier than that of the frequency, at the point where the valve is approximately 70% valve closure closed. This is closer to the start of the transition period than the result of the offline analysis, and as such is more useful for real-time response. The inferred behaviour models also closely follow the observed data.

7.1.3 Industrial Application

Chapter 6 presents the validation of the scheduling and FD-HM algorithms, as well as a discussion about the requirements for industrial integration and implementation. It concludes by presenting a scheduling and FD-HM approach that combines the described contributions.

Modular Production Line Scheduling Algorithm

Factory production times for a batch of products were anonymised and the MILP algorithm was used to generate a schedule for them. This was then compared to a schedule generated using an identical product batch and a simplified version of the factory scheduler. The results were similar to those presented in Chapter 4, with reductions in the cost functions for the single and parallel machine setups and the serial setup failing to find an improved schedule in the allotted time. The reductions do not appear as significant because of the comparatively shorter setup times relative to the overall processing times, but the reduction in total processing time directly correlates to costs incurred during manufacturing processes and as such any reduction is significant when considered in the context of high-volume production. The production line simulation shows an improvement in the cost function of 15% and a 28% increase in the longest delay, as per the results of Chapter 4.

Fault Detection and Health Monitoring

The wave soldering machine use case was further described alongside details about the initial attempts to collect data for analysis. Magnet-mounted accelerometers collected half-hourly pump vibration samples over the course of a month. This was shown to be too infrequent a sampling rate, and thus impossible to analyse effectively without supplementary data from the production and maintenance schedules, which could not be accessed at the time. The pump power level is dependant upon the products being passed over the solder wave and without a production schedule it is impossible to match up times when the pump is operating at specific power levels. As such, a number of artificial data sets were generated, each one representative of a pump response to a particular condition while operating at a set power level. This included: a response proportional to the flow rate behaviour shown in the test bench; a response proportional to the rate of pipe blockage; a step change in behaviour and finally a temporary anomaly.

The offline BCD analysis successfully locates the change points in most of the artificial data sets, and extracts the parameters that generated them. The quadratic response data set highlighted the sensitivity of BCD to the choice of priors used in the analysis. It also demonstrated the flexibility they allow for meeting the requirements of the analysis. In the case of the temporary anomaly response data set, the BCD analysis filtered out the anomaly, which is potentially a useful result dependant on the machine being monitored. However, it was shown that by adapting the algorithm to detect additional change points or splitting the data set would allow the analysis to successfully locate the anomalous behaviour.

The OBCD analysis is similarly successful in locating change point in most of the artificial data sets, and infers behaviour models that closely follow the actual models. It is less effective in locating a change point in the quadratic response data set, where it segments the behaviour about the centre and shows

a high uncertainty about the location of the change point. This is a result of the way that the OBCD algorithm infers a linear behaviour model based on incoming data. The analysis of the temporary anomaly data set filters out the anomalous behaviour when observed at the end of the time period, but does locate the start of the anomaly as it happens in real time. Both behaviours are potentially valuable to users.

Implementation and Integration

Next, this chapter presented the planned implementation of the FD-HM method in an industrial environment and integration with a local Industrial Internet of Things (IIoT) network. The benefits of such integration include an improved ability to monitor machine health through the combination of the real time data analysis and the expected machine behaviour extrapolated from the production schedule. These results could then be disseminated to user devices quickly and efficiently to provide information about current machine health and potential maintenance requirements. Additionally, storage of this data could be used to supplement the information stored in existing databases or Product Lifecycle Management (PLM) software, or to facilitate the creation of a historical database for further analysis in the future.

Combined Scheduling/FD-HM Approach

Finally, a combined scheduling/FD-HM approach was discussed, taking advantage of the modular nature of the MILP algorithms and the increased knowledge of machine state offered by the BCD analyses. The scheduling system would create an optimised schedule based on informed decisions about current machine health and upcoming maintenance events. Faults resulting in unplanned machine downtime would be detected quickly, and users would be notified through their devices. Updated local schedules could quickly be generated as part of a rapid response and recovery protocol using the scheduling

algorithms designed for the local machine group setups. This would provide the necessary time required for an updated global schedule to be generated and implemented.

7.2 Future work

This section will briefly outline the potential work that can be done to fully implement the contributions of the thesis. Firstly, the extension and implementation of the scheduling algorithm will be addressed. Secondly, additional assessments of the offline and online BCD algorithms will be discussed with the aim of refining them to better suit the requirements of the use case. Finally, the steps towards full integration of the combined approach will be discussed.

Modular Production Line Scheduling Algorithm

While the performance of the MILP scheduling algorithm was shown to generally be better than that of the PRBM by most metrics, it suffers from long processing times due to its complexity. The time limit imposed to reflect time pressures in industrial environments limits the ability of the solver to complete the optimisation. This problem would certainly be exacerbated by the increased number of product types present in an existing factory, as well as the increased number of machines. This creates the opportunity to refine the scheduling algorithm to make it more efficient. It could also be an effective way to take advantage of the elastic computing capabilities offered by cloud technology. Additionally, further work is needed to establish how effectively the algorithm could be integrated with an existing Manufacturing Execution System (MES) and the best way to implement automatic local re-scheduling in the case of unplanned machine failures.

Fault Detection and Health Monitoring

Additional assessment and refinement of the presented BCD methods is possible. One of the key future goals is to complete the analysis of the industrial data collected at a higher sample rate. This, combined with the supplementary production data described in 6, would significantly emphasise the value of these methods. This information would have allowed the data to be filtered such that only readings from a single pump power level were present for analysis. Alternatively, the readings could have been normalised with respect to the pump power level, potentially creating a larger, more cohesive data set and reducing the risk of missing features indicative of a fault. Alongside this, there is scope to further test the BCD methods for false alarms, missed detection and delayed detection. These tests would be particularly important for the on-line BCD method. Knowledge about these potential weaknesses would allow for further refinement and development of the algorithms in use. A secondary goal is the validation of the assumptions made for the artificial data sets and their representations of potential machine faults. As this FD-HM method is intended to provide a generic solution to various factory machine, additional fault behaviours will have to be considered dependant on the machines being analysed.

Implementation and Combined Approach

Finally, full implementation and assessment of the discussed combined scheduling and FD-HM approach would be a valuable topic to pursue. Such an implementation would represent a significant step towards the digital transformation of a company. A quantitative assessment of the effective improvement to a factory would encourage further practical research and could take place on an advanced factory simulation or as part of a small scale trial on a production line.

Reference List

- [1] C. Y. Liu and S. C. Chang, "Scheduling flexible flow shops with sequence-dependent setup effects," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 4, pp. 408–419, 2000.
- [2] A. Allahverdi, J. N. D. Gupta, and T. Aldowaisan, "A review of scheduling research involving setup considerations," *The International Journal of Management Science*, vol. 27, 1999.
- [3] A. Allahverdi and H. M. Soroush, "The significance of reducing setup times/setup costs," *European Journal of Operational Research*, vol. 187, no. 3, pp. 978–984, 2008.
- [4] A. Allahverdi, "The third comprehensive survey on scheduling problems with setup times / costs," *European Journal of Operational Research*, vol. 246, no. 2, pp. 345–378, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.ejor.2015.04.004>
- [5] Jonny Williamson, "How did digital technology turn a suburban Cheshire factory into a world-beater," 6 2019.
- [6] Joe Bush, "Making an old factory ready for a new era of manufacturing," 8 2022.
- [7] J. Manyika, J. Sinclair, R. Dobbs, G. Strube, L. Rassey, J. Mischke, J. Remes, C. Roxburgh, K. George, D. O'Halloran, and S. Ramaswamy, "Manufacturing the future: The next era of global growth and innovation The McKinsey Global Institute," *McKinsey Global Institute, McKinsey Operations Practice*, 2012.
- [8] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *NIST Special Publication 800-145*, 2011.
- [9] M. Weiser, "The computer for the 21st century," *Scientific American, Volume 265*, pp. 94–104, 9 1991. [Online]. Available: <https://www.scientificamerican.com/article/the-computer-for-the-21st-century/>
- [10] ITU, "The Internet of Things," *Itu Internet Report 2005*, p. 212, 2005.
- [11] J. Voas, "Networks of ' Things '," *NIST Special Publication 800-183*, 2016.
- [12] J. Conway, "The Industrial Internet of Things: An Evolution to a Smart Manufacturing Enterprise," *White Paper, Schneider Electric*, 2016.

- [13] GE Digital, “Predix: The Platform for the Industrial Internet,” *GE Digital White Paper*.
- [14] D. Zuehlke, “SmartFactory - From vision to reality in factory technologies,” in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2008.
- [15] —, “SmartFactory-Towards a factory-of-things,” in *Annual Reviews in Control*, 2010.
- [16] European Commission, “Factories of the Future PPP: Towards Competitive EU Manufacturing,” *European Commission*, pp. 1–2, 2013. [Online]. Available: http://ec.europa.eu/research/press/2013/pdf/ppp/fof_factsheet.pdf
- [17] W. Wahlster, J. Helbig, A. Hellinger, M. A. V. Stumpf, J. Blasco, H. Galloway, and H. Gestaltung, “Recommendations for implementing the strategic initiative INDUSTRIE 4.0,” 2013.
- [18] T. G. K. Simonis, Y-S Gloy, “Automation in weft knitting technology,” *IOP Conference Series: Materials Science and Engineering*, vol. Volume 141.
- [19] N. Jazdi, “Cyber physical systems in the context of Industry 4.0,” *2014 IEEE Automation, Quality and Testing, Robotics*, pp. 2–4, 2014. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6857843
- [20] IEC, “Factory of the Future,” *IEC White Paper*, 2015. [Online]. Available: <http://www.iec.ch/whitepaper/futurefactory/?ref=extfooter>
- [21] W. Bauer, M. Hämmerle, S. Schlund, and C. Vocke, “Transforming to a Hyper-connected Society and Economy – Towards an “Industry 4.0”,” *Procedia Manufacturing*, vol. 3, pp. 417–424, 2015.
- [22] S. Kalpakjian, S. R. Schmid, U. Teknologi, M. Prentice, H. Singapore, L. New, Y. Toronto, S. Tokyo, M. Mexico, C. Munich, P. Capetown, and H. Kong Montreal, *Manufacturing engineering and technology*, 6th ed. Prentice Hall, 2006.
- [23] D. Lei and J. D. Goldhar, “Computer-integrated Manufacturing (CIM): Redefining the }Manufacturing Firm into a Global Service Business,” *International Journal of Operations & Production Management*, vol. 11, no. 10, pp. 5–18, 1 1991. [Online]. Available: <https://doi.org/10.1108/EUM0000000001288>
- [24] T. J. Kramer, J. T. Chibnall, and B. D. Pedersen, “Managing Computer Integrated Manufacturing (CIM): A Review of Themes, Issues, and Practices,” *Tech. Rep. 4*, 1992. [Online]. Available: <https://about.jstor.org/terms>
- [25] D. Kolberg and D. Zühlke, “Lean Automation enabled by Industry 4.0 Technologies,” in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2015.

- [26] K. Ejsmont, B. Gladysz, and A. Kluczek, “Impact of industry 4.0 on sustainability-bibliometric literature review,” 7 2020.
- [27] UK Government, “Made Smarter Review 2017,” p. 231, 2017. [Online]. Available: <https://www.gov.uk/government/publications/made-smarter-review>
- [28] VID/VDE, “Reference Architecture Model Industrie 4.0 (RAMI4.0),” *Igarss 2014*, no. 1, p. 28, 2015.
- [29] IIC, “The Industrial Internet of Things Volume G1: Reference Architecture,” *Iic:Pub:G1:V1.80:20170131*, 2017. [Online]. Available: https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf
- [30] S. S. V. Keshav Kolla, D. M. Lourenço, A. A. Kumar, and P. Plapper, “Retrofitting of legacy machines in the context of Industrial Internet of Things (IIoT),” in *Procedia Computer Science*, vol. 200. Elsevier B.V., 2022, pp. 62–70.
- [31] M. J. McNally, J. C. Chaplin, G. Martínez-Arellano, and S. Ratchev, “Data Capture and Visualisation on a Shoestring: Demonstrating the Digital Manufacturing on a Shoestring Project,” *IFIP Advances in Information and Communication Technology*, vol. 620 IFIP, pp. 200–209, 2021.
- [32] S. Mittal, M. A. Khan, D. Romero, and T. Wuest, “A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs),” pp. 194–214, 10 2018.
- [33] D. McFarlane, “Digital Manufacturing on a Shoestring,” 2022. [Online]. Available: www.digitalshoestring.net
- [34] Talend, “Definitive Guide To Data Integration,” Talend, Tech. Rep., 2019.
- [35] Lars Fielder, “Why customer analytics matter — McKinsey,” McKinsey&Company, Tech. Rep. May, 2016. [Online]. Available: <https://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/why-customer-analytics-matter>
- [36] Talend, “What are Data Silos?” 2021. [Online]. Available: <https://www.talend.com/resources/what-are-data-silos/>
- [37] —, “Definitive Guide Cloud DataWarehouses And Cloud Data Lakes Contents,” Talend, Tech. Rep.
- [38] J. Patel, “Bridging Data Silos Using Big Data Integration,” *International Journal of Database Management Systems*, vol. 11, no. 3, pp. 01–06, 2019.
- [39] F. Van den Abeele, J. Hoebeke, I. Moerman, and P. Demeester, “Integration of Heterogeneous Devices and Communication Models via the Cloud in the Constrained Internet of Things,” *International Journal*

- of *Distributed Sensor Networks*, vol. 2015, pp. 1–16, 2015. [Online]. Available: <http://www.hindawi.com/journals/ijdsn/2015/683425/>
- [40] M. Díaz, C. Martín, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing,” *Journal of Network and Computer Applications*, vol. 67, pp. 99–117, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.jnca.2016.01.010>
- [41] Amazon, “Amazon Web Services,” 2017. [Online]. Available: <https://aws.amazon.com/>
- [42] Microsoft, “Microsoft Azure,” 2017. [Online]. Available: <https://azure.microsoft.com/en-gb/>
- [43] —, “Microsoft OneDrive,” 2017. [Online]. Available: <https://onedrive.live.com/about/en-gb/>
- [44] L. Leong, “Critical Capabilities for Public Cloud Infrastructure as a Service, Worldwide,” *Gartner*, 2015. [Online]. Available: <https://www.gartner.com/doc/reprints?id=1-2QQX6UM&ct=151027&st=sb>
- [45] Salesforce, “Salesforce,” 2017. [Online]. Available: <https://www.salesforce.com/uk/>
- [46] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [47] Siemens, “Mindsphere,” 2017. [Online]. Available: <http://www.siemens.com/global/en/home/company/topic-areas/digitalization/mindsphere.html>
- [48] S. V. Novikov and A. A. Sazonov, “Application of the open operating system ‘MindSphere’ in digital transformation of high-tech enterprises,” *Economics Journal*, vol. 1, no. 1, pp. 20–26, 2019.
- [49] P. Mosca, Y. Zhang, Z. Xiao, and Y. Wang, “Cloud Security: Services, Risks, and a Case Study on Amazon Cloud Services,” *International Journal of Communications, Network and System Sciences*, vol. 7, no. 12, pp. 529–535, 2014. [Online]. Available: <https://login.ctu.idm/oclc.org/?url=http://search.proquest.com/docview/1645562992?accountid=26967>
- [50] BBC, “Hack attack causes ‘massive damage’ at steel works,” 2014. [Online]. Available: <http://www.bbc.co.uk/news/technology-30575104>
- [51] B. Grobauer, T. Walloschek, and E. Stöcker, “Understanding cloud computing vulnerabilities,” *IEEE Security and Privacy*, vol. 9, no. 2, pp. 50–57, 2011.
- [52] N. Woolf, “DDoS attack that disrupted internet was largest of its kind in history, experts say,” San Francisco, 10 2016. [Online]. Available: <https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet>

- [53] I. Stojmenovic and S. Wen, “The Fog Computing Paradigm: Scenarios and Security Issues,” *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, vol. 2, pp. 1–8, 2014. [Online]. Available: <https://fedcsis.org/proceedings/2014/drp/503.html>
- [54] R. Roman, J. Zhou, and J. Lopez, “On the features and challenges of security and privacy in distributed internet of things,” *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.comnet.2012.12.018>
- [55] R. Roman, P. Najera, and J. Lopez, “Securing the Internet of Things,” *Computer*, vol. 44, no. 9, pp. 51–58, 9 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/6017172/>
- [56] A. Varghese and D. Tandur, “Wireless requirements and challenges in Industry 4.0,” in *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, 2014.
- [57] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus, and M. Fallgren, “Scenarios for 5G mobile and wireless communications: The vision of the METIS project,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.
- [58] B. Placzek, “Selective data collection in vehicular networks for traffic control applications,” *Transportation Research Part C: Emerging Technologies*, vol. 23, pp. 14–28, 2012.
- [59] M. Sepulcre, J. Gozalvez, and B. Coll-Perales, “Multipath QoS-driven routing protocol for industrial wireless networks,” *Journal of Network and Computer Applications*, 2016.
- [60] M. Aazam and E. N. Huh, “Fog computing and smart gateway based communication for cloud of things,” *Proceedings - 2014 International Conference on Future Internet of Things and Cloud, FiCloud 2014*, pp. 464–470, 2014.
- [61] C. Rhodes, “Manufacturing : statistics and policy,” *House of Commons Briefing Paper*, no. 01942, p. 12, 2015.
- [62] A. Allahverdi, C. T. Ng, T. C. Cheng, and M. Y. Kovalyov, “A survey of scheduling problems with setup times or costs,” *European Journal of Operational Research*, vol. 187, no. 3, pp. 985–1032, 2008.
- [63] W. J. Hopp and M. L. Spearman, *Factory Physics*, third edit ed. New York: McGraw-Hill, 2008.
- [64] C. Artigues, P. Baptiste, G. Bel, J.-C. Billaut, C. Bloch, H. Camus, J. Carlier, J.-B. Cavallé, D. Costa, D. Dubios, P. Esquirol, H. Fargier, P. Fortemps, J.-C. Gentina, A. Hertz, M.-J. Huguet, O. Korbaa, P. Lopez, M.-A. Manier, E. Neron, A. Oliver, M.-C. Portmann,

- C. Prins, F. Roubellat, C. Varnier, A. Vignier, and M. Widmer, *Production Scheduling*, first edit ed., P. Lopez and F. Roubellat, Eds. London: ISTE, 2001.
- [65] M. L. Pinedo, *Scheduling*, 5th ed. Cham: Springer International Publishing, 2016. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-26580-3>
- [66] S. French, *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, 1st ed. Chichester: Ellis Horwood, 1982.
- [67] J. M. Framinam, R. Leisten, and R. R. García, *Manufacturing Scheduling Systems*, 1st ed., Springer, Ed. London: Springer-Verlag Berlin Heidelberg, 2014.
- [68] J. R. Jackson, "Scheduling a production line to minimize maximum tardiness," *Research Report, Management Science Research Project*, 1955.
- [69] A. Otto and C. Otto, "How to design effective priority rules: Example of simple assembly line balancing," *Computers and Industrial Engineering*, vol. 69, no. 1, pp. 43–52, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.cie.2013.12.013>
- [70] I. Sabuncuoglu, "A study of scheduling rules of flexible manufacturing systems: A simulation approach," *International Journal of Production Research*, vol. 36, no. 2, pp. 527–546, 1998. [Online]. Available: <http://dx.doi.org/10.1080/002075498193877>
- [71] W. Shen and D. H. Norrie, "Agent-Based Systems for Intelligent Manufacturing : A State-of-the-Art Survey," *Knowledge and Information Systems*, vol. 1, no. 2, pp. 129–156, 1999.
- [72] T. N. Wong, C. W. Leung, K. L. Mak, and R. Y. K. Fung, "Dynamic shopfloor scheduling in multi-agent manufacturing systems," *Expert Systems with Applications*, vol. 31, no. 3, pp. 486–494, 2006.
- [73] J. Reaidy, P. Massotte, and D. Diep, "Comparison of negotiation protocols in dynamic agent-based manufacturing systems," *International Journal of Production Economics*, vol. 99, no. 1-2, pp. 117–130, 2006.
- [74] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.
- [75] R. G. Smith, "The Contract Net Protocol: High-level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, 1980.
- [76] N. K. Krothapalli and a. Deshmukh, "Design of negotiation protocols for multi-agent manufacturing systems," *International Journal of Production Research*, vol. 37, no. October, pp. 1601–1624, 1999.

- [77] A. Gordillo and A. Giret, “Performance Evaluation of Bidding-Based Multi-Agent Scheduling Algorithms for Manufacturing Systems,” *Machines*, pp. 233–254, 2014.
- [78] P. Gu, S. Balasubramanian, and D. H. Norrie, “Bidding-based process planning and scheduling in a multi-agent system,” *Computers & Industrial Engineering*, vol. 32, no. 2, pp. 477–496, 1997.
- [79] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, “Towards smart factory for Industry 4.0: A self-organized multi-agent system with big data based feedback and coordination,” *Computer Networks*, vol. 101, pp. 158–168, 2015.
- [80] F. P. Maturana and D. H. Norrie, “Multi-agent Mediator architecture for distributed manufacturing,” *Journal of Intelligent Manufacturing*, vol. 7, pp. 257–270, 1996.
- [81] W. S. Gere Jr, “A Heuristic Approach to Job Shop Scheduling,” Ph.D. dissertation, Carnegie Institute of Technology, 1962.
- [82] J. L. Snowdon, S. Bhattacharyya, and G. J. Koehler, “A Review of Machine Learning in Scheduling,” *IEEE Transactions on Engineering Management*, vol. 41, no. 2, pp. 165–171, 1994.
- [83] J. A. Blessing and B. A. Watford, “INFMSS, an Intelligent FMS Scheduling System,” *World Productivity Forum and 1987 Ann. Int. Industrial Engineering Conference*, pp. 82–88, 1987.
- [84] M. J. Shaw, S. C. Park, and N. Raman, “Intelligent Scheduling with Machine Learning Capabilities: The Induction of Scheduling Knowledge,” 3 1990. [Online]. Available: <https://www.ideals.illinois.edu/bitstream/handle/2142/29867/intelligentsched1639shaw.pdf?sequence=2>
- [85] R. Mántaras, “Case-Based Reasoning,” in *Machine Learning and Its Applications*, 2001, pp. 127–145.
- [86] G. Chrysosolouris, M. Lee, and M. Domroese, “The use of neural networks in determining operational policies for manufacturing systems,” *Journal of Manufacturing Systems*, vol. 10, pp. 166–175, 1991.
- [87] J. Xu, “Improved Genetic Algorithm to Solve the Scheduling Problem of College English Courses,” *Complexity*, vol. 2021, 2021.
- [88] Y. YIH and A. THESEN, “Semi-Markov decision models for real-time scheduling,” *International Journal of Production Research*, vol. 29, no. 11, pp. 2331–2346, 1991. [Online]. Available: <https://doi.org/10.1080/00207549108948086>
- [89] A. S. Manne, “On the Job-Shop Scheduling Problem,” *Operations Research*, vol. 8, pp. 219–223, 1960.
- [90] J. Błażewicz, W. Domschke, and E. Pesch, “The job shop scheduling problem: Conventional and new solution techniques,” *European Journal of Operational Research*, vol. 93, 1996.

- [91] Y. Unlu and S. J. Mason, "Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems," *Computers & Industrial Engineering*, vol. 58, no. 4, pp. 785–800, 2010. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0360835210000483>
- [92] Z. Li and M. Ierapetritou, "Process scheduling under uncertainty: Review and challenges," *Computers and Chemical Engineering*, vol. 32, no. 4-5, pp. 715–727, 2008.
- [93] J. Smith and Z. Taskin, "A Tutorial Guide to Mixed Integer Programming Models and Solution Techniques," *Optimization in Medicine and Biology*, pp. 1–23, 2008. [Online]. Available: http://www.ie.boun.edu.tr/~taskin/pdf/IP_tutorial.pdf
- [94] S. Hazelhurst, "Scientific computing using virtual high-performance computing: a case study using the Amazon Elastic Computing Cloud," *SAICSIT '08*, pp. 94–103, 2008. [Online]. Available: <http://doi.acm.org/10.1145/nnnnnnn.nnnnnn>
- [95] M. Queyranne and A. S. Schulz, "Polyhedral approaches to machine scheduling," *Technical Report*, pp. 1–62, 1994. [Online]. Available: <http://en.scientificcommons.org/47857053>
- [96] J. P. Sousa and L. A. Wolsey, "A time indexed formulation of non-preemptive single machine scheduling problems," *Mathematical Programming*, vol. 54, no. 1-3, pp. 353–367, 2 1992. [Online]. Available: <http://link.springer.com/10.1007/BF01586059>
- [97] D. Houck, J. C. Picard, M. Queyranne, and R. R. Vemuganti, "The traveling salesman problem and shortest n-paths," *University of Maryland, College Park*, 1977.
- [98] E. Cakici and S. J. Mason, "Parallel machine scheduling subject to auxiliary resource constraints," *Production Planning and Control*, vol. 18, no. 3, pp. 217–225, 2007.
- [99] K. Kempf, R. Uzsoy, S. Smith, and K. Gary, "Evaluation and comparison of production schedules," *Computers in Industry*, vol. 42, no. 2, pp. 203–220, 2000.
- [100] G. P. Georgiadis, A. P. Elekidis, and M. C. Georgiadis, "Optimization-based scheduling for the process industries: From theory to real-life industrial applications," *Processes*, vol. 7, no. 7, 2019.
- [101] S. J. Honkomp, S. Lombardo, O. Rosen, and J. F. Pekny, "The curse of reality - Why process scheduling optimization problems are difficult in practice," *Computers and Chemical Engineering*, vol. 24, no. 2-7, pp. 323–328, 2000.
- [102] J. L. Loveland, S. K. Monkman, and D. J. Morrice, "Dell uses a new production-scheduling algorithm to accommodate increased product variety," *Interfaces*, vol. 37, no. 3, pp. 209–219, 2007.

- [103] S. C. Trovinger and R. E. Bohn, "Setup Time Reduction for Electronics Assembly: Combining Simple (SMED) and IT-Based Methods," *Production and Operations Management*, vol. 14, no. 2, pp. 205–217, 2009. [Online]. Available: <http://doi.wiley.com/10.1111/j.1937-5956.2005.tb00019.x>
- [104] C. A. Méndez and J. Cerdá, "An MILP-based approach to the short-term scheduling of make-and-pack continuous production plants," *OR Spectrum*, vol. 24, no. 4, pp. 403–429, 2002.
- [105] T. Westerlund, J. Isaksson, and I. Harjunkski, "Solving a two-dimensional trim-loss problem with MILP," *European Journal of Operational Research*, vol. 104, no. 3, pp. 572–581, 1998.
- [106] P. M. Castro, I. Harjunkski, and I. E. Grossmann, "New continuous-time scheduling formulation for continuous plants under variable electricity cost," *Industrial and Engineering Chemistry Research*, vol. 48, no. 14, pp. 6701–6714, 2009.
- [107] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, 1st ed., ser. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2001. [Online]. Available: <http://link.springer.com/10.1007/978-1-4471-0347-9>
- [108] V. T. Do and U. P. Chong, "Signal model-based fault detection and diagnosis for induction motors using features of vibration signal in two-dimension domain," *Strojniski Vestnik/Journal of Mechanical Engineering*, vol. 57, no. 9, pp. 655–666, 2011.
- [109] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, "Human-machine-interaction in the industry 4.0 era," in *Proceedings - 2014 12th IEEE International Conference on Industrial Informatics, INDIN 2014*, 2014.
- [110] J. Schimert, "Data-Driven Fault Detection Based on Process Monitoring using Dimension Reduction Techniques," in *2008 IEEE Aerospace Conference*. IEEE, 3 2008, pp. 1–12. [Online]. Available: <http://ieeexplore.ieee.org/document/4526621/>
- [111] S. Xu, "A Survey of Knowledge-Based Intelligent Fault Diagnosis Techniques," *Journal of Physics: Conference Series*, vol. 1187, no. 3, 2019.
- [112] A. Angelopoulos, E. T. Michailidis, N. Nomikos, P. Trakadas, A. Hatziefremidis, S. Voliotis, and T. Zahariadis, "Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects," 1 2020.
- [113] M. Prist, A. Monteriù, A. Freddi, P. Cicconi, F. Giuggioloni, E. Ciazer, C. Verdini, and S. Longhi, "Online Fault Detection: a Smart Approach for Industry 4.0," *IEEE International Workshop on Metrology for Industry 4.0 and IoT : virtual conference, proceedings : June 3-5, 2020.*, 6 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9138295/authors#authors>

- [114] H. Webert, T. Döb, L. Kaupp, and S. Simons, “Fault Handling in Industry 4.0: Definition, Process and Applications,” *Sensors*, vol. 22, no. 6, 3 2022.
- [115] Micro-Epsilon, “Measuring the solder-wave height on wave soldering machines,” Micro-Epsilon, Birkenhead, Tech. Rep., 2022.
- [116] —, “Micro-Epsilon Product Guide,” Ortenburg, 2022.
- [117] Solderstar, “Solderstar,” 2022. [Online]. Available: <https://www.solderstar.com/en/>
- [118] C. Scheffer, “Pump Condition Monitoring Through Vibration Analysis,” *PUMPS: Maintenance, Design, and Reliability Conference 2008 – IDC Technologies*, pp. 1–20, 2008.
- [119] R. Birajdar, R. Patil, and K. Khanzode, “Vibration and noise in centrifugal pump - sources and diagnosis methods,” *3rd International Conference on Integrity, Reliability and Failure*, no. July, pp. 20–24, 2009. [Online]. Available: https://www.researchgate.net/publication/265543012_VIBRATION_AND_NOISE_IN_CENTRIFUGAL_PUMPS_SOURCES_AND_DIAGNOSIS_METHODS/download
- [120] D. L. Rivera, M. R. Scholz, M. Fritscher, M. Krauss, and K. Schilling, “Towards a Predictive Maintenance System of a Hydraulic Pump,” *16th IFAC Symposium on Information Control Problems in Manufacturing*, 2018.
- [121] D. Siano and M. A. Panza, “Diagnostic method by using vibration analysis for pump fault detection,” *Energy Procedia*, vol. 148, no. Ati, pp. 10–17, 2018. [Online]. Available: <https://doi.org/10.1016/j.egypro.2018.08.013>
- [122] J. Stojek, “Application of Time-Frequency Analysis for Diagnostics of Valve Plate Wear in Axial-Piston Pump,” *Archive of Mechanical Engineering*, vol. 57, no. 3, pp. 309–322, 2010.
- [123] T. Shinn, R. Carpenter, and R. C. Fales, “State Estimation Techniques for Axial Piston Pump Health Monitoring,” in *ASME/BATH 2015 Symposium on Fluid Power and Motion Control*. American Society of Mechanical Engineers, 10 2015. [Online]. Available: <https://asmedigitalcollection.asme.org/FPST/proceedings/FPMC2015/57236/Chicago,%20Illinois,%20USA/289537>
- [124] A. Dutta, I. A. Karimi, and S. Farooq, “PROAD (Process Advisor): A health monitoring framework for centrifugal pumps,” *Computers and Chemical Engineering*, vol. 163, 7 2022.
- [125] Y. Qin, W. T. Li, C. Yuen, W. Tushar, and T. K. Saha, “IIoT-Enabled Health Monitoring for Integrated Heat Pump System Using Mixture Slow Feature Analysis,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4725–4736, 7 2022. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9416833>

- [126] S. A. Kim, Y.I. and M. t. Lambert, "Behavior of orifices and blockages in unsteady pipe flows," *9th Annual Symposium on Water Distribution Systems Analysis*, no. January 2007, pp. 15–19, 2007.
- [127] S. A. COLLET and M. KÄLLMAN, *Pipe Vibrations Measurement*. Energiforsk, 2017. [Online]. Available: <https://energiforskmedia.blob.core.windows.net/media/22212/pipe-vibrations-energiforskrappport-2017-351.pdf>
- [128] N. L. T. Lile, M. H. M. Jaafar, M. R. Roslan, and M. S. M. Azmi, "Blockage Detection in Circular Pipe Using Vibration Analysis," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 2, no. 3, p. 252, 2016.
- [129] S. Aminikhanghahi, T. Wang, and D. J. Cook, "Real-time change point detection with application to smart home time series data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 1010–1023, 2019.
- [130] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 339–367, 2017.
- [131] D. Rindskopf, "Overview of Bayesian Statistics," *Evaluation Review*, vol. 44, no. 4, pp. 225–237, 2020.
- [132] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M. G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, and C. Yau, "Bayesian statistics and modelling," *Nature Reviews Methods Primers*, vol. 1, no. 1, 2021.
- [133] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, Eds., *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 5 2011. [Online]. Available: <https://www.taylorfrancis.com/books/9781420079425>
- [134] E. Ruggieri and M. Antonellis, "An exact approach to Bayesian sequential change point detection," *Computational Statistics and Data Analysis*, vol. 97, pp. 71–86, 5 2016.
- [135] R. P. Adams and D. J. C. MacKay, "Bayesian Online Changepoint Detection," *Technical Report, University of Cambridge*, 2007. [Online]. Available: <http://arxiv.org/abs/0710.3742>
- [136] P. Fearnhead and Z. Liu, "On-line inference for multiple changepoint problems," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 589–605, 9 2007. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2007.00601.x>
- [137] A. Mohammad-Djafari and O. Féron, "A Bayesian approach to change points detection in time series," *International Journal of Imaging Systems and Technology*, vol. 16, no. 5, pp. 215–221, 2006.

- [138] C. Davidson-Pilon, *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference (Addison-Wesley Data and Analytics)*, 1st ed. Addison-Wesley Professional, 2015. [Online]. Available: <https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers>
- [139] E. Ruggieri, “A Bayesian approach to detecting change points in climatic records,” *International Journal of Climatology*, vol. 33, no. 2, pp. 520–528, 2013.
- [140] O. Niculita, Z. Skaf, and I. K. Jennions, “The application of Bayesian change point detection in UAV fuel systems,” in *Procedia CIRP*, vol. 22, no. 1. Elsevier B.V., 2014, pp. 115–121. [Online]. Available: <http://dx.doi.org/10.1016/j.procir.2014.07.119>
- [141] R. Malladi, G. P. Kalamangalam, and B. Aazhang, “Online Bayesian change point detection algorithms for segmentation of epileptic activity,” in *Asilomar Conference on Signals, Systems and Computers*, no. November 2013, 2013, pp. 1833–1837.
- [142] E. Tochev, H. Pfifer, and S. Ratchev, “A Comparison of Centralised and Decentralised Scheduling Methods Using a Simple Benchmark System,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1287–1292, 2018. [Online]. Available: <https://doi.org/10.1016/j.ifacol.2018.08.355>
- [143] J. Jiménez, Gómez, M. D. Buhrmester, A. Vázquez, H. Whitehouse, and W. B. Swann, “The Dynamic Identity Fusion Index: A New Continuous Measure of Identity Fusion for Web-Based Questionnaires,” *Social Science Computer Review*, vol. 34, no. 2, pp. 215–228, 2016.
- [144] Brüel & Kjær, “Measuring vibration,” *White paper*, p. 42, 1982. [Online]. Available: <https://www.bksv.com/media/doc/br0094.pdf>
- [145] L. A. Wolsey, *Integer Programming*, 1st ed. New York, NY: John Wiley & Sons, Inc., 1998. [Online]. Available: https://ostad.nit.ac.ir/upload/Integer_Programming_1.pdf
- [146] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer Programming*, ser. Graduate Texts in Mathematics. Cham: Springer International Publishing, 2014, vol. 271. [Online]. Available: <http://solab.kaist.ac.kr/files/IP/IP2017/2014.IntegerProg-Conforti-Cornuejols-Zambelli.pdf><http://link.springer.com/10.1007/978-3-319-11008-0>
- [147] R. C. Jeroslow, “There Cannot be any Algorithm for Integer Programming with Quadratic Constraints,” *Operations Research*, vol. 21, no. 1, pp. 221–224, 1973. [Online]. Available: <https://www.jstor.org/stable/pdf/169101.pdf?refreqid=excelsior%3A2752e5c1b8fa61bb13d025d3911f69b3>
- [148] G. Dantzig, *Linear Programming and Extensions*. RAND Corporation, 1963. [Online]. Available: <https://www.rand.org/pubs/reports/R366.html>

- [149] E. D. Anderson, J. Gondzio, C. Mészáros, and X. Xu, “Implementation of Interior-Point Methods for Large Scale Linear Programs,” University of Geneva, Geneva, Tech. Rep. February, 1 1996.
- [150] T. Illés and T. Terlaky, “Pivot versus interior point methods: Pros and cons,” *European Journal of Operational Research*, vol. 140, no. 2, pp. 170–190, 2002.
- [151] A. H. Land and A. G. Doig, “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica*, vol. 28, no. 3, p. 497, 7 1960. [Online]. Available: <https://www.jstor.org/stable/1910129?origin=crossref>
- [152] T. Achterberg, “Constraint Integer Programming,” Ph.D. dissertation, Technischen Universität Berlin, Berlin, 7 2008.
- [153] M. Fischetti and A. Lodi, “Heuristics in Mixed Integer Programming,” *Wiley Encyclopedia of Operations Research and Management Science*, pp. 1–6, 2011.
- [154] R. E. Gomory, “Outline of an Algorithm for Integer Solutions to Linear Programs,” *Bulletin of the American Mathematical Society*, vol. 64, no. 5, pp. 275–278, 1958.
- [155] S. . J. G . Dantzig, R . Fulkerson, “Solution of a Large-Scale Traveling-Salesman Problem,” *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954. [Online]. Available: <http://www.jstor.org/stable/166695>
- [156] C.-Y. R. Huang, C.-Y. Lai, and K.-T. T. Cheng, “CHAPTER 4 - Fundamentals of algorithms,” in *Electronic Design Automation*, L.-T. Wang, Y.-W. Chang, and K.-T. T. Cheng, Eds. Boston: Morgan Kaufmann, 2009, pp. 173–234. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780123743640500114>
- [157] J. E. Mitchell, “Branch-and-Cut Algorithms for Combinatorial Optimization Problems,” *Handbook of Applied Optimization*, pp. 65–77, 2002.
- [158] M. Padberg and G. Rinaldi, “Optimization of a 532-city symmetric traveling salesman problem by branch and cut,” *Operations Research Letters*, vol. 6, no. 1, pp. 1–7, 1987.
- [159] —, “A Branch-and-Cut Approach to a Traveling Salesman Problem with Side Constraints,” *Management Science*, vol. 35, no. 11, pp. 1393–1412, 1989.
- [160] J. Löfberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, pp. 284–289, 2004.
- [161] “Gurobi Optimizer,” 2021. [Online]. Available: <https://www.gurobi.com/products/gurobi-optimizer/>

- [162] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, D. B. Rubin, J. Carlin, H. Stern, D. Rubin, and D. Dunson, *Bayesian Data Analysis Third edition (with errors fixed as of 6 April 2021)*, 2021, no. April.
- [163] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*. Massachusetts: Athena Scientific, 2008. [Online]. Available: [https://ece307.cankaya.edu.tr/uploads/files/introductiontoprobability\(bertsekas,2nd,2008\).pdf](https://ece307.cankaya.edu.tr/uploads/files/introductiontoprobability(bertsekas,2nd,2008).pdf)
- [164] J. J. K. Ó Ruanaidh and W. J. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing*, ser. Statistics and Computing. New York, NY: Springer New York, 1996. [Online]. Available: <http://link.springer.com/10.1007/978-1-4612-0717-7>
- [165] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [166] W. K. Hastings, “Monte Carlo Sampling Methods Using Markov Chains and Their Applications Published,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970. [Online]. Available: <https://academic.oup.com/biomet/article-abstract/57/1/97/284580>
- [167] P. J. Green, “Reversible jump Markov chain monte carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.
- [168] W. F. Darnieder, “Bayesian methods for data-dependent priors,” Ph.D. dissertation, The Ohio State University, 2011. [Online]. Available: http://rave.ohiolink.edu/etdc/view?acc_num=osu1306344172
- [169] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, “Hybrid Monte Carlo,” *Physics Letters B*, vol. 195, no. 2, pp. 216–222, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/037026938791197X>
- [170] M. D. Hoffman and A. Gelman, “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo,” *Journal of Machine Learning Research*, vol. 15, pp. 1593–1623, 2014. [Online]. Available: <https://jmlr.org/papers/volume15/hoffman14a/hoffman14a.pdf>
- [171] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984.
- [172] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov chain Monte Carlo methods,” *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 72, no. 3, pp. 269–342, 2010.

- [173] F. Liang and W. H. Wong, “Evolutionary Monte Carlo: Applications to Cp model sampling and change point problem,” *Statistica Sinica*, vol. 10, no. 2, pp. 317–342, 2000.
- [174] J. S. Liu and R. Chen, “Sequential monte carlo methods for dynamic systems,” *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.
- [175] S. A. Sisson, Y. Fan, and M. A. Beaumont, “Overview of Approximate Bayesian Computation,” *Handbook of Approximate Bayesian Computation*, no. 1, pp. 1–66, 2 2018. [Online]. Available: <http://arxiv.org/abs/1802.09720>
- [176] H. Rue, S. Martino, and N. Chopin, “Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations,” *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 71, no. 2, pp. 319–392, 2009.
- [177] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational Inference: A Review for Statisticians,” *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017. [Online]. Available: <https://doi.org/10.1080/01621459.2017.1285773>
- [178] R. van de Schoot, M. Miočević, and (Eds.), *Small Sample Size Solutions: A Guide for Applied Researchers and Practitioners*, 1st ed., R. van de Schoot and M. Miočević, Eds. London: Routledge, 2 2020. [Online]. Available: <https://www.taylorfrancis.com/books/9781000760941>
- [179] E. Tochev, D. Rengasamy, H. Pfifer, and S. Ratchev, “System condition monitoring through Bayesian change point detection using pump vibrations,” in *IEEE International Conference on Automation Science and Engineering*, vol. 2020-Augus, 2020, pp. 667–672.
- [180] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, “Probabilistic programming in Python using PyMC3,” *PeerJ Computer Science*, vol. 2016, no. 4, p. e55, 4 2016. [Online]. Available: <https://peerj.com/articles/cs-55>
- [181] R. M. Neal, “Slice sampling,” *Annals of Statistics*, vol. 31, no. 3, pp. 705–767, 2003.
- [182] D. Barry and J. A. Hartigan, “A Bayesian Analysis for Change Point Problems,” *Journal of the American Statistical Association*, vol. 88, no. 421, p. 309, 1993.
- [183] G. J. J. v. d. Burg and C. K. I. Williams, “An Evaluation of Change Point Detection Algorithms,” pp. 1–33, 2020. [Online]. Available: <http://arxiv.org/abs/2003.06222>
- [184] R. C. Wilson, M. R. Nassar, and J. I. Gold, “Bayesian online learning of the hazard rate in change-point problems,” *Neural Computation*, vol. 22, no. 9, pp. 2452–2476, 2010.

- [185] E. Tochev, H. Pfifer, and S. Ratchev, “Indirect System Condition Monitoring Using Online Bayesian Change-point Detection,” in *Smart Technologies for Precision Assembly*, S. Ratchev, Ed. Cham: Springer International Publishing, 2021, pp. 81–92.
- [186] S. W. Smith, “Digital Signal Processing, Chapter 15,” in *California Technical Publishing*, 1999, pp. 277–284. [Online]. Available: https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_Ch15.pdf
- [187] A. Gelman and D. B. Rubin, “Inference from Iterative Simulation Using Multiple Sequences,” *Statistical Science*, vol. 7, no. 7, pp. 354–457, 1992. [Online]. Available: <https://projecteuclid.org/euclid.ss/1177011136>
- [188] Alpha Assembly Solutions, *Wave Soldering Troubleshooting Guide*. Surrey: Alpha Assembly Solutions, 2016.
- [189] C. Shea, “Simple, Effective Process Control in Wave Soldering,” Siemens Information and Communication Networks, Inc., Tech. Rep., 6 1999. [Online]. Available: https://smtnet.com/library/index.cfm?fuseaction=view_article&article_id=924&company_id=50692
- [190] C. E. Shannon, “Communication in the Presence of Noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1 1949. [Online]. Available: <https://doi.org/10.1109/jrproc.1949.232969>
- [191] Siemens, “Siemens Teamcenter,” 2022. [Online]. Available: <https://www.plm.automation.siemens.com/global/en/products/teamcenter/>
- [192] —, “Siemens SIMATIC IOT2050,” 2021. [Online]. Available: <https://new.siemens.com/global/en/products/automation/pc-based/iot-gateways/simatic-iot2050.html>
- [193] K. GmBH, “Revolution Pi,” 2021. [Online]. Available: <https://revolutionpi.com/>
- [194] Siemens, “MindConnect Nano,” 2021. [Online]. Available: <https://www.dex.siemens.com/mindsphere/mindconnect/MindConnect-Nano>