Sebastian Dransfeld

Measurement and Supervision in Automated Production

Thesis for the degree of doktor ingeniør

Trondheim, July 2007

Norwegian University of
Science and Technology
Faculty of Engineering Science & Technology
Department of Production & Quality Engineering

NTNU
Innovation and Creativity

NTNU
Norwegian University of Science and Technology

Thesis for the degree of doktor ingeniør

Faculty of Engineering Science & Technology
Department of Production & Quality Engineering

ISBN 978-82-471-2940-1 (printed ver.)
ISBN 978-82-471-2954-8 (electronic ver.)
ISSN 1503-8181

Theses at NTNU, 2007:134

Printed by Tapir Uttrykk

Acknowledgments

The work on this thesis was carried out between 2002 and 2006 as a part of the P2005 program. P2005 is a program funded by the Norwegian Research Council, and is aimed at improving and increasing productivity in Norway. The work was conducted at the Department of Production and Quality Engineering at the Norwegian University for Science and Technology (NTNU), and was supervised by Professor Terje K. Lien.

During the work I have participated in two projects at SINTEF (The Foundation for Scientific and Industrial Research), one in co-operation with Raufoss United (now Kongsberg Automotive - Raufoss Works), and the other in co-operation with Volvo Aero Norge AS. In both projects the aim was to increase productivity by introducing new manufacturing methods, and add supervision of the processes. Due to these projects and good co-operation from the participating companies I gained valuable experience and a good foundation for this thesis.

A special thanks goes to Lars Erik Wetterwald for supporting me during the complete work on this thesis, and pushing me forward when I was going nowhere, and to my dear Therese for her support and inspiration during the whole period.

Contents

Acknowledgments iii

Table of Contents v

Abstract xi

1 Introduction 1

1.1 Background 1

1.2 The next level 5

1.3 Knowledge 6

1.4 Overall Equipment Efficiency 6

1.5 Examples 8

2 State-of-the-Art 11

2.1 Introduction 11

2.2 Intelligent and Flexible Assembly Systems 12

2.2.1 Introduction 12

2.2.2 Requirements 12

2.2.3 Sensors 13

2.2.4 Computer vision 14

2.2.5 Tactile sensors 14

2.2.6 Force and Torque sensors 14

2.2.7 Research 15

2.3 Tool Condition Monitoring (TCM) 16

2.3.1 Introduction 16

2.3.2 Tool wear 16

2.3.3 Sensors for TCM 17

2.3.4 Research 20

2.3.5 Industrial Applications 21

I Logging of Events in Automated Production 23

3 Infrastructure of Automated Production 25

3.1 Background 25

3.2 Logical controllers 25

3.3 Industrial networks 26

3.4 Human Machine Interface 27

3.5 Supervisory Control and Data Acquisition 28

3.6 OLE for Process Control 28

3.6.1 OPC interface details 29

3.6.2 Shortcomings of OPC 29

3.7 Overall Equipment Efficiency 30

3.7.1 Systems for OEE 31

4 Data Collection 33

4.1 Background 33

4.2 Data storage 34
4.3 Network technology 34
4.3.1 Protocols 35
4.3.2 Transmission efficiency 36
4.4 Real-Time 36
4.5 How to collect data 38
4.5.1 Reliable timestamps 39
4.5.2 Event ordering 40

5 Discrete event logging 41
5.1 Introduction 41
5.2 Simulation 42
5.3 Production lines 42
5.4 First Experiment 43
5.4.1 Introduction 43
5.4.2 Robot cell 45
5.4.3 Assembly setup 46
5.4.4 Data collection 48
5.4.5 Results 50
5.4.6 Impact on system 55
5.4.7 Conclusions 55
5.5 Second Experiment 56
5.5.1 Introduction 56
5.5.2 Assembly cell 56
5.5.3 Data collection 57

5.5.4 Results 58

5.5.5 Impact on system 62

5.5.6 Conclusions 62

6 Summary 64

6.1 Introduction 64

6.2 Personnel 64

6.3 Engineers needs 65

6.4 Operator needs 66

II Force and Vibration Measurements for Error Detection 67

7 Tool Condition Monitoring 69

7.1 Introduction 69

7.2 Laboratory equipment 71

7.3 Experiments 72

7.3.1 Tools 72

7.3.2 Machining 73

7.3.3 Measurements 73

7.4 Results 74

7.4.1 Tool wear 74

7.4.2 Computations 75

7.5 Tool condition monitoring (TCM) 79

7.6 Discussion 82

7.7 Conclusions 83

8 Force Measurement in Pick and Place Operations 84

8.1 Introduction 84

8.1.1 Background 84

8.1.2 Testing strategies 85

8.1.3 Force measurement as a way of testing 85

8.2 Equipment 86

8.3 Experiments 87

8.3.1 First experiment 87

8.3.2 Second experiment 88

8.4 Force measurements 88

8.5 Results 90

8.5.1 First experiment 90

8.5.2 Second experiment 91

8.6 Applicability 93

8.6.1 Corrective action 93

8.6.2 Optimization of assembly operation 95

8.7 Conclusion 96

9 Summary 97

9.1 Methodology for force measurements in assembly operations 97

9.2 Initial development 97

9.3 Tuning 98

9.4 Quality control 98

9.5 Adaptive assembly 98

III User Interface 101

10 User Interface 103

10.1 Introduction 103

10.2 Spreadsheets 104

10.3 Web interface 106

10.4 Conclusions 107

Bibliography 108

Thesis advisor
Professor Terje K. Lien

Author
Sebastian Dransfeld

Measurement and supervision in automated production

Abstract

This thesis consists of two main parts.

Part I:

A general introduction to data collection and storage. Experiments are conducted to store events with timestamps in two different production cells. The information is used to increase the productivity by finding the inefficient parts of the production. The information is also used to get a greater overview of the different parts of the production to create better production equipment configurations.

Part II:

Two very different cases are studied to find out how increased knowledge of the process can improve the processes.

The first case is to monitor forces and vibration in milling. Experiments are conducted in milling of a difficult-to-cut material, and some methods are proposed on how a worn out tool can be detected.

The second case is to monitor forces in an assembly operation. Experiments are conducted to find out how the forces gathered can be used to increase the knowledge about the assembly operations, and some methods are proposed on how the information can be used to increase productivity and quality.

Chapter 1

Introduction

1.1 Background

There has been three main events which increased the ability of man to produce more with less labour. The first is the industrial revolution. The industrial revolution introduced machines that did labour intensive work. The second was the introduction of standardisation and mass production. A main example here is the T-Ford. Because of how Henry Ford designed and produced the T-Ford it became cheap and available for the masses. The third and last event was the introduction of automation and computer technology. Automation has made it possible to create factories where no hard labour has to be done.

This is a dream come true for man. But not completely. Today we have a possibility of producing goods without doing the hard labour, but to realise this potential we need to do a lot more work on efficiency, productivity and accuracy.

The first automation equipment was implemented in the automotive industry in the 50s and the 60s. The initial use was to do simple repetitive tasks, often handling heavy or bulky parts. Productivity was increased and production went up. As the world had a great hunger for products, there was no problem to sell everything that was produced. So the main goal of the automation was to increase productivity and production.

In the 70s and the 80s information technology (IT) entered the arena. IT made it possible to create more advanced automation equipment, it became possible to automate more complex tasks. But the tasks still had to be repetitive. Automated manufacturing lead to a considerable increase in world production capacity, and as more countries from the developed world wanted a part of the prosperity which the developed world experienced, the increase in production became so large that there no longer was any guarantee of selling the products.

The world changed from a producers market to a consumers market. The producers had to fulfill the needs of the consumers (or what the commercials told the consumers they needed). So automation became important not just to increase productivity, but also to improve quality and reduce costs.

During the 90s and 00s this trend has continued. In addition transport has become cheap and borders are easier to cross. It is now possible to produce goods on one side of the planet, and sell it on the other. Multinational companies with pressure from their shareholders to earn even more money, rather move production to a cheap labour country than to invest in automation equipment. Unless the automation equipment results in a better product at lower cost.

Money is not the only objective for actions, there are other causes to strive for. Automation equipment should be developed for more complex tasks, and not exploit cheap labour where current technology fails. Today fish is sent from Norway to China for handling, before it is returned to Europe. The reasons are the high cost of Norwegian labour, but also that the work is heavy and unpleasant. So instead of solving this problem by automation, the problem is just moved elsewhere. The result is reduced value creation, and increased use of transport. The first part damages the economy, the latter part may be an environmental problem.

So automation today is not just about making good products at low labour cost. Automation has the potential of maintaining production in the developed world so that we can continue our lifestyle. As the worlds population grows older, automation also has the potential of doing tasks not just in the manufacturing industry, but also in healthcare [53] where there wont be enough caretakers to move elderly around and doing exercises. And automation may contribute to solve a lot of other problems the world faces.

This thesis will look at one of the tools which will advance manufacturing automation to the next level. The tool is termed Computer-integrated manufacturing (CIM). Efficient processing and transportation of information through all manufacturing processes in a factory, and through the processes needed for introduction of new products, has to be integrated. CIM will increase utilization of current automation equipment and enable us to solve more complex tasks in the future. CIM must not be realised as one big program to do it all, but as several different programs which can communicate through standard interfaces and by a network.

The figure 1.1 is a common CIM concept model. It is developed by Siemens and is discussed in "Computer Integrated Manufacturing and Engineering" [64, pages 53–55]. The model dates to 1989, and since then many parts of the model has been completed with its computer systems and integration.

One essential information flow is the flow from product design to product manufacturing. Designers today use computer programs to create all parts and features. The finished product is split into separate parts, and then transferred to a program which translates the part information into cutter location data. The cutter location data is a machine independent generic format which describes what material has to be removed from a workpiece to create the actual part. If the part is to be machined, the cutter location data is then transformed into ISO-code. The ISO-code is loaded into a machining centre, which compiles the ISO-code into native machine language.

For planning, sales purchasing etc. there are several large systems to handle this. SAP is one of the best known and largest companies in this field. These systems are autonomous and most engineers wont have much to do with them. But it is vital that these systems are integrated with the production information to know what has been produced, what the stock is, and what is to be produced.

One piece of the CIM model which has not gotten too much attention is the flow of detailed information from the shop floor to the engineers. There seems to be two camps of programmers, one camp which programs computers, and one camp which programs manufacturing equipment. These two camps have not been able to communicate too well (literally and physically), but this is changing.

This thesis will look into what information is necessary to collect from different parts of a production process, and how it can be made accessible. The information can be collected at several levels, and different types of information are of interest at different stages of a production facilities life-cycle.

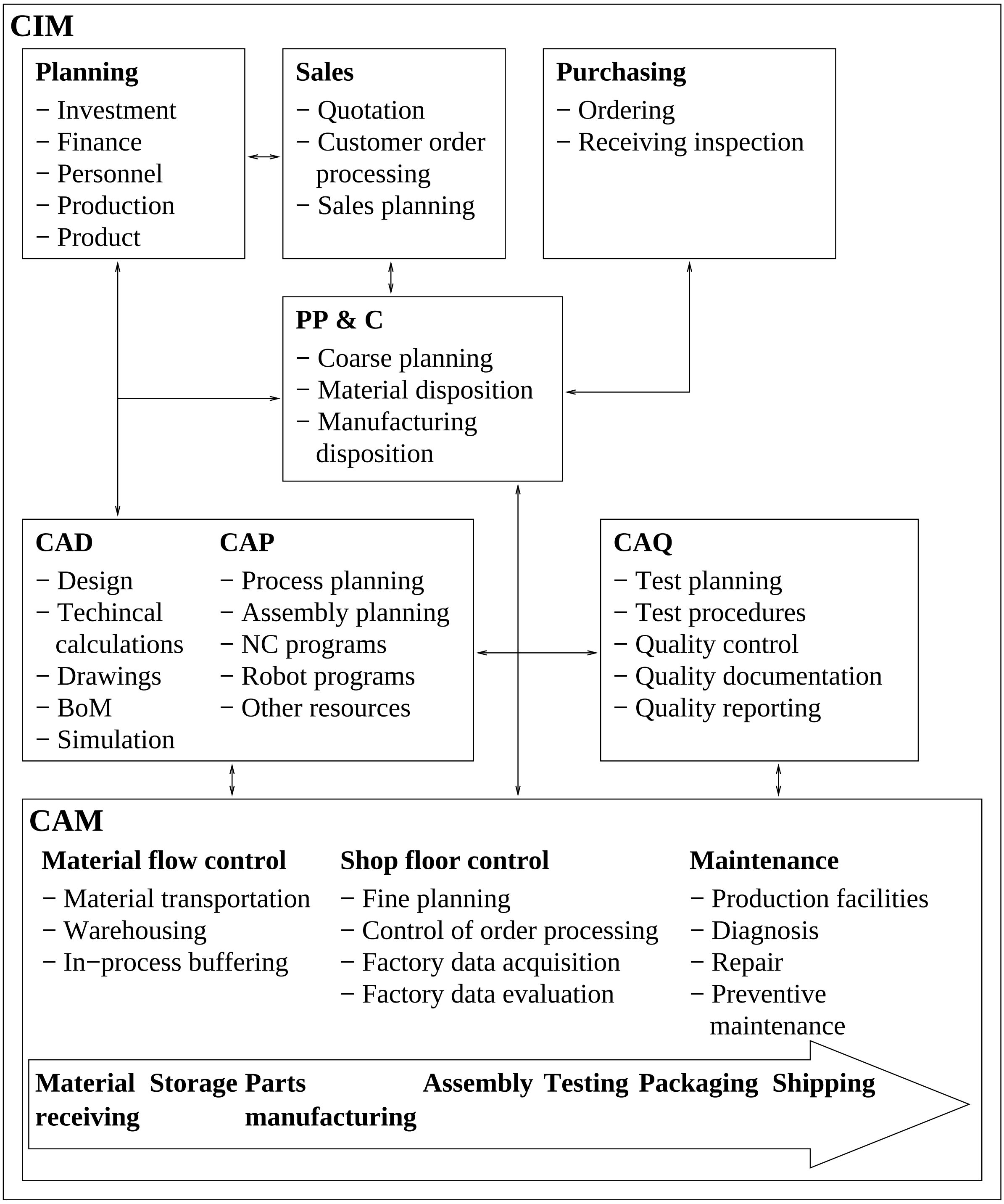


Figure 1.1: Computer-Integrated Manufacturing [64, page 54]

1.2 The next level

As mentioned in the previous section information technology is the tool to advance industrial production to the next level.

The problem which needs to be solved is to be able to manufacture and assemble parts with higher complexity and in smaller volumes. Todays solutions for automation are expensive and not very flexible. To be cost effective a large production volume is required and the production has to be continuous, as illustrated by the automotive industry. The automotive industry today boasts a large variety of models and options, BMW claims that no car in the 7-series is the same, but all the customisation and flexibility is created by having a set of standard parts which can be assembled in a variety of ways. And the assembly is done with human labour.

So the next generation of manufacturing and assembly systems should be able to handle a large variety of components, and to be able to adjust quickly between production of different products.

And of course there is always a constant demand for increased productivity and less errors.

For automation of complex tasks the equipment needs some knowledge about the elements that are to be handled in the process. Human beings, with their five senses and high mobility, can perform very complex tasks, and they can easily adapt knowledge if there are some small changes in the process to be done.

To make this work there has to be done a lot of research in a lot of different areas, but one thing is a common factor: Knowledge. And this is where the information technology must do its magic.

Information must be collected at several different levels in each operation. The information must be stored for further evaluation. Some results from the evaluation will go straight back into the process to adapt it, some results will go to the operators so that they can increase their knowledge, some results will go to the engineers so that they can improve the current system and to invent better solutions for new systems, and some information will go to the administration to inform them on what is going on in the production.

So there are many who need information, and what information they need will vary. It is possible to extract and aggregate detailed information to a higher level, but the opposite is impossible. So it should be strived for a high level of information detail to be able to fulfill all information needs.

1.3 Knowledge

Information collected from a production system can be stored and analysed on three levels.

The first level is the data level. Raw data can be collected from a large variety of sensors, and there are several methods for importing and storing these data onto a computer system to make them available.

But raw data alone are of no use, as a large set of numbers do not tell anything. So data has to be transformed to the next level, "information". Information is data set into system. The data needs metadata. The metadata describes what the data represents, and how to interpret them. Other useful metadata is how the data was captured and who did it. Creating metadata is immensely important, as it has to describe the data so well that no information which cannot be recreated later is lost.

When the information is secured, it can be transformed to the third level which is knowledge. The information has to be analysed to find out what they tell us about a system, and what conclusions can be drawn from them. Two types of information can be birth rate and death rate. If the information is interpreted, we can see that if the birth rate is lower than the deathrate, the population will sink. If force data from a machining centre is interpreted, and there is a large peak in forces when the tool enters the workpiece, the feed rate is to high when entering the workpiece. This is simple knowledge.

After information is secured and transformed into knowledge it is important to feed the knowledge back into the manufacturing system. This is the tough part, since a manufacturing system will never be as flexible and intelligent as a human. But there is a lot of improvement potential from the current non-intelligent robot without any sensing and adaptability.

1.4 Overall Equipment Efficiency

Much can be done with increased knowledge about manufacturing processes, but the umbrella most of this thesis belongs under is the search for increased Overall Equipment Efficiency (OEE). OEE is a benchmark figure based on Availability, Performance and Quality. This approach will result in a better benchmark then using uptime and throughput.

OEE as a tool comes from the production philosophy lean manufacturing. Lean manufacturing is based on the work Toyota has done to improve their production. Key lean manufacturing principles include [46]:

- Pull processing – products are pulled from the consumer end, not pushed from the production end
- Perfect first-time quality – quest for zero defects, revealing & solving problems at the source
- Waste minimization – eliminating all activities that do not add value & safety nets, maximize use of scarce resources (capital, people and land)
- Continuous improvement – reducing costs, improving quality, increasing productivity and information sharing
- Flexibility – producing different mixes or greater diversity of products quickly, without sacrificing efficiency at lower volumes of production
- Building and maintaining a long term relationship with suppliers through collaborative risk sharing, cost sharing and information sharing arrangements.

> Lean is basically all about getting the right things, to the right place, at the right time, in the right quantity while minimizing waste and being flexible and open to change. [46]

As mentioned earlier OEE is based on three elements. There is much that can influence the OEE, but some typical factors are [62]:

Availability	Planned downtime Set up time Unplanned recorded downtime or breakdowns
Performance	Reduced speed Minor unrecorded stoppages
Quality	Rejects Rework Yield and Start up losses

The work done in this thesis touches many of these points, points that can be benchmarked by OEE. The first part of my thesis is focused on logging of events in a production line. The data can be analysed to figure out where unnecessary time is wasted. With figures and numbers it is easier to make decisions on what problems should be prioritised and if investments in improvements will be cost effective.

The second part of my thesis is focused on improving process quality. Increased process quality is important to reduce scrap and to improve the product itself. Both by reducing the amount of parts with errors which are shipped to a customer and to get a better finish.

Without a focus on integrating information collection, and working on increased knowledge about processes, the benefits from "thinking lean" will be reduced. Lean manufacturing needs a well defined feedback from the production system to show its full potential.

In addition will increased knowledge of processes lead the way to more intelligent and flexible solutions. This is both a key principle in lean manufacturing and a necessity to increase the amount of tasks done in the world which can be automated.

1.5 Examples

Some factories have begun the work to integrate information from the shop floor into the rest of the facility.

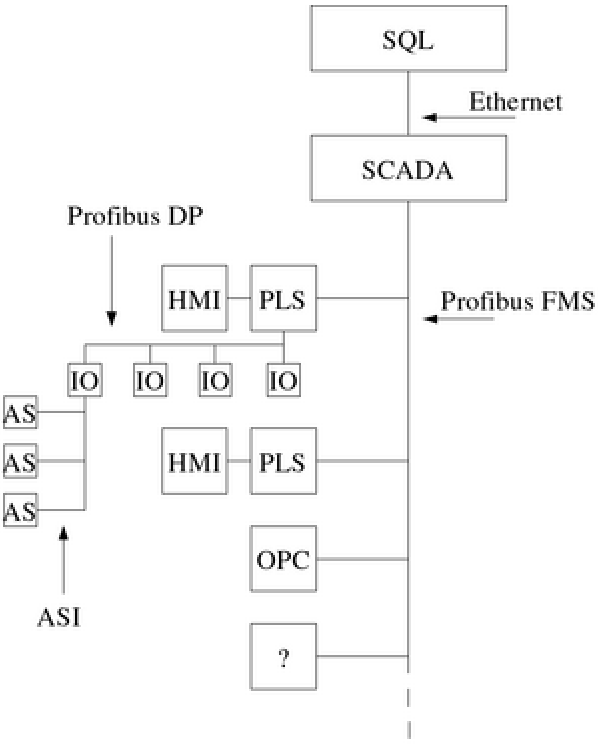


Figure 1.2: Raufoss Technology network

At Raufoss Technology the integration of information was one area of focus when the new production line was constructed in 2000. Figure 1.2 shows the main design principle for the information network. The main goal for this information network was to gather quality data, so that the production processes could be documented for the customer.

The production line consists of three main parts. The first part is metal forming, the second is machining, and the third is assembly.

From the first part critical parameters from the metal forming, primarily temperature and time, is collected. These parameters are used to adjust the process over time, no direct control with each part. There are additional checks to control if a part is faulty, but these measurements are not gathered in the main system.

The machining equipment is difficult to communicate with, so no information is gathered here. The problem is that the control unit of the machining centres are closed and do not speak the same "language" as the data acquisition system.

From the assembly line fault information is gathered for each part, so that it is possible to connect types of failures with relevant parts. This makes it possible to pinpoint on which stations improvements have to be made.

After the production line was operational, it was analysed to find areas which could be improved to increase total output. People who worked in the production had some ideas of how some processes could be improved, but no one knew exactly how the ideas would affected the output. To find the improvement potential of the different ideas, the data collection system was analysed. It was then discovered that the information level of detail was too low, and it was impossible to improve it. By bypassing the data collection system, and fetching data directly from some PLCs it was possible to get data with a good enough level of detail. As an example it was discovered that one additional buffer station improved the average cycle-time in the assembly line by 15%.

The data collection system is now configured to create shift-reports where number of parts and quality is recorded. This system works well on a day to day basis, but it is still not possible to use it for in-depth analysis.

Another factory which has incorporated a data collection system into their production facilities is ELKO. They have added a data collection system to several already existing production lines. The data collection system works almost like the system at Raufoss Technology.

Critical points in the production line, primarily feeding of parts, are identified. When these systems fail, the error and duration is stored. This makes it possible to analyse the efficiency and throughput of the system, but it is not possible to do in-depth analysis.

These two examples show that work has been done on data collection from production lines, and that the information can be used. But the level of information is to shallow to do in-depth analysis, and therefore the information cannot be used to really improve the lines. Problems can be detected, but often the problem source or the impact of the problem remains hidden.

Other examples are "Implementation of TPM in cellular manufacture" [27], where the authors have used data collection and OEE measurements to decide how TPM was to be planned, and "From overall equipment efficiency (OEE) to overall Fab effectiveness (OFE)" [60], where the authors describe the necessity of having a complete view of the factory when working with data acquisition and OEE measurements.

Chapter 2

State-of-the-Art

2.1 Introduction

The work done in this thesis is focused on improving manufacturing processes by collection data from them, so the knowledge of the inner workings can be increased. There are a lot of manufacturing processes which can benifit from data collection, but it is not possible to cover all.

In this thesis two main cases has been studied: assembly and machining. In both cases the work done is twofold: The first is to find solutions which can improve the processes, the second is two create a setup which allows data to be collected and analysed so that the solutions found can be qualified. This chapter will present the state-of-the-art in flexible assembly and tool condition monitoring.

Assembly and machining are very different processes, but when looking from monitoring perspective there are similarities.

For example can the sensors used for collecting information be the same. In this thesis force sensors are used to both check tool health and an assembly process.

The objective of the different cases are also the same: To increase OEE. From improving low level processes with the intent of improving quality to higher level monitoring to detect problem areas which should be adressed.

2.2 Intelligent and Flexible Assembly Systems

Mass production and automated systems has increased production efficiency a lot. To improve efficiency in the future more complex assembly systems has to be developed.

One reason for this is that complex processes which require all human senses and the human brain to be performed, has to be automated. Current automated systems are still non-intelligent, and this must be improved.

Another reason for more intelligent and flexible assembly systems is to be able to do more work with fewer standardised manufacturing lines. One production line for one product will not work in the future, one production line must be used for a large variety of products.

2.2.1 Introduction

Automation has been one important reason for increased productivity in modern manufacturing. Until the 80s assembly automation was equal to dedicated assembly machines. Dedicated assembly lines require high volume products to be cost effective [68].

Today, globalization has created a huge excess production capacity. This has changed the market condition from demand exceeding supply, to supply exceeding demand. This has created a situation where the consumers can demand products which fit their needs and taste [43].

To be able to supply such a market, mass customization is necessary. In principle the customer should be able to get a unique product. In automated manufacturing this is difficult to realize, so the solution is to create a large set of standard parts which can be assembled to a large variety of products.

To solve assembly needs for mass customized products, there is a need for flexible automation systems which can change quickly between assembling different product [34], and there is a need for intelligent assembly systems which can cope with varieties within one product [34].

2.2.2 Requirements

A lot of systems today are marketed as flexible, but often they are not used as intended. It is possible to use a set of flexible automation components and put them together to a non-flexible system. Often there are only small changes which must be done to such systems to exploit their possibilities.

The following are some characteristics which are required for a flexible assembly system today [58]:

- Ability to make quick product changes – adaptability
- Easy programmable and reprogrammable – trainable
- Minimal product specific tooling – dexterous
- Resilience to manufacturing tolerances – compliant and skilful
- Error detection and recovery – observant and meticulous

2.2.3 Sensors

The importance of sensors in assembly systems was stressed in a CIRP keynote paper in 1978 [77]. Since then a lot of research has been done on sensors, and sensors in assembly has become commonplace [68]. To create a flexible and intelligent manufacturing system without sensors is impossible, as sensors is the link between the real world and the control logic within the manipulators.

There are a lot of different sensors on the market today, with different qualities and possibilities. This section will focus on the main type of sensors and their use in assembly.

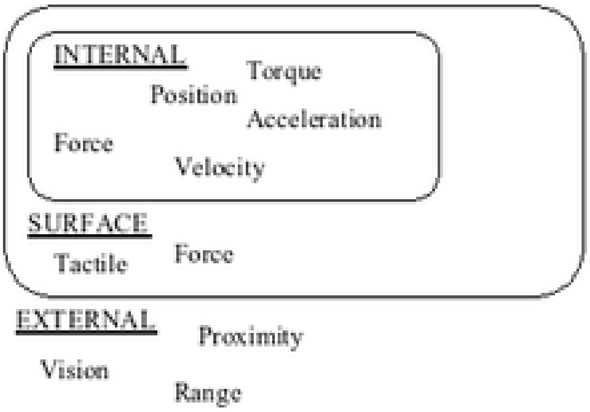


Figure 2.1: Sensors in robotics

2.2.4 Computer vision

The basics of computer vision is to take a picture of the environment, split the picture into pixels (picture points), and extract information from it to understand what the contents of the picture is.

Today the most common use of computer vision is to find already known products at an unknown location within a defined area, and in automatic inspection. These tasks are almost always done in 2D.

There also exist small cameras which can be used to look where humans cannot go, like inside the human body.

There has been done much research on extending computer vision to be able to see in 3D, and to understand not previously known objects. This research has not been reflected in practical use, were the focus is on refining simple tasks.

2.2.5 Tactile sensors

Tactile sensors are used to measure if there is contact between two objects. Tactile sensors can supply the distribution of contact over a surface, and the best tactile sensors are more sensitive than the human fingertip.

These sensors can be used for simple tasks like detecting whether a part has been picked up, to more delicate tasks like execution of high precision insertions.

2.2.6 Force and Torque sensors

Force and torque sensors have been researched extensively. The main usage for force and torque sensors is to make robots interact better with their environment. Standard robots have only knowledge about space, so it is difficult for them to handle object which are soft or have undefined positions.

Another focus area for force sensing is in interaction with humans. Robots have powerful motors and will easily hurt humans which come into their work envelope. If robots are to be used together with humans, either in a co-operative work environment or in health care, it must be made sure that the robots will not hurt anyone.

2.2.7 Research

Holonic Manufacturing Systems (HMS)

A holon is defined by the HMS consortium as "an autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects" [74]. In practice this means that a manufacturing cell should not be designed for a specific task, but it should provide a service like drilling, welding etc.

A Holonic Manufacturing System is "a holarchy which integrates the entire range of manufacturing activities from order booking through design, production and marketing to realize the agile manufacturing enterprise".

Research on HMS is therefore conducted on many levels, from shop-floor [39], via abstract production [45, 47], to enterprise [37].

The conclusion to most of the research is that it is simple in theory to decide that the manufacturing system should provide a service, but in practice most manufacturing systems in use today are to static and to tied up to one specific task.

Human Machine Systems (HMS)

Since manufacturing systems today are far from intelligent, there will be need for human labor in factories for many years to come. So instead of finding fully automated solutions for all tasks, many researchers work on finding solutions where machines assist humans, instead of replacing them. Such systems are often named Human Machine Systems.

HMS can either be implemented as a direct assistant for Humans [44], or as a co-worker which shares the workload [36].

Robot Control

A standard robot is non-intelligent and has no good sense of space. A lot of research has been done to improve this.

One important area to improve setup and reconfiguration of robots is off-line programming. But when programming off-line real world coordinates has to be used. If the robot is not 100% correctly aligned to the world coordinate system which the off-line programming is done in, work has to be done to synchronize this. One method for calibrating is to use vision technology [78].

Another important aspect to improve robot behavior is to make the robot interact better with its surroundings. One area of research is to make robots able to interact with large and non-rigid objects [38], whilst others use force control to be able to interact with objects without defined positions [20].

Improvement of robot control walks hand in hand with increased computational power, and is in some cases suitable for practical use.

Industrial applications

Although Intelligent and Flexible Assembly Systems has been researched on for many years, there are still not many industrial cases. One problem might be that research is creating solutions which are too complex for practical industrial use. Today it is impossible to invest money in a manufacturing system which can not be proved reliable. Most companies rather move production to a low-cost country then to invest in unproven technology.

2.3 Tool Condition Monitoring (TCM)

A lot of manufacturing today is done by cutting metal. If the processes used are improved this will have a major impact on productivity in the worlds industry.

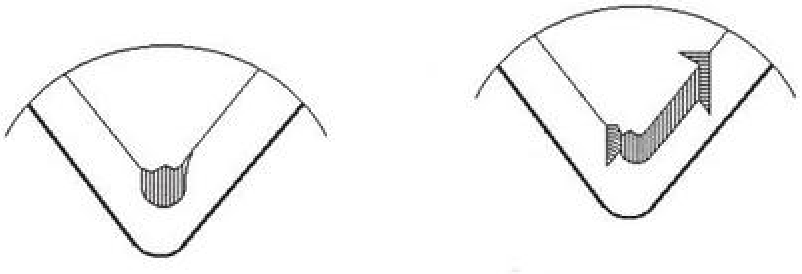
2.3.1 Introduction

In manufacturing there is a constant drive for cost savings and performance improvements. This requires factories with minimal manning. Untended machining requires monitoring to detect anomalies and to assure that operations are working as planned. In addition there is a need to improve tool usage to reduce wear and to prolong tool life. The use of sensor systems to monitor tool condition in machining is becoming more commonplace to enhance productivity.

2.3.2 Tool wear

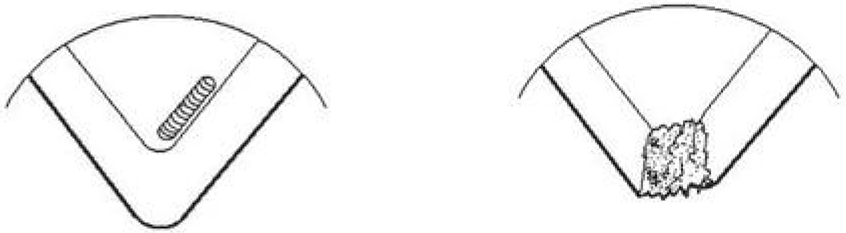
Tool wear can be classified into several types and summarized as follows [31]:

- Adhesive wear associated with shear plane deformation.
- Abrasive wear resulting from hard particles cutting action.
- Diffusion wear occurring at high temperatures.
- Fracture wear such as chipping due to fatigue.



(a) Nose wear (low cutting speed)

(b) Flank and notch wear
(medium cutting speed)



(c) Cratering (high cutting speed)

(d) Plastic/breakage
(very high cutting speed)

Figure 2.2: Cutting tool wear forms in orthogonal cutting.

In figure 2.2 [31] the predominant wear modes for turning operation is shown. The source of the different wear modes is a mixture of the different wear types.

2.3.3 Sensors for TCM

A sensor system for TCM can be either continuous or intermittent, and it can measure directly or indirectly. In a continuous system the measured variable(s) will be available throughout the process, whilst in an intermittent system it will only be available at predefined intervals. A direct measurement system will find and provide the actual value of the measured variable, for example flank wear; an indirect measurement system will measure a property which can be correlated to the actual value. An optimal system would be a continuous system with direct measurement. This might not always be feasible.

Detection

TCM can be used to detect various incidents. In machining it can be used to detect tool breakage, tool wear and chatter. In grinding it can be used to detect chatter and dressing points. In addition, collisions and missing tools and workpieces can be detected, and it is desired that the quality of the workpiece can be determined.

To be able to fulfill these tasks, TCM must often be implemented as an in-process system. This is because tool wear is a complex phenomenon that manifests itself in different and varied ways.

By implementing TCM the objective is to insure optimum system performance, and should therefore serve the following purposes [31]:

- Advanced fault detection system for cutting and machine tool.
- Check and safeguard machining process stability.
- Maintain machining tolerance by compensating for tool wear.
- Machine tool damage avoidance system.

Requirements

Since TCM often is implemented in-process, sensors for TCM must fulfill these requirements [73]:

- Measurement as close to the machining point as possible.
- No reduction in the static and dynamic stiffness of the machine tool.
- No restriction of working space and cutting parameters.
- Wear- and maintenance-free, easily changed with low cost.
- Resistant to dirt, chips and cooling liquid.
- Resistant to mechanical, electromagnetic and thermal influences.
- Function independent of tool or workpiece.
- Adequate metrological characteristics.
- Reliable signal transmission.

Force

It has been widely established that there is a correlation between tool wear and cutting force [73, 31]. Many commercial systems today are based on force sensors [21, 25].

The best force sensors today are based on piezoelectric sensors. Piezoelectric sensors have a high resolution, and are very small. They can therefore be implemented in the workpiece fixture or in the tool spindle.

There has been a thorough research on the use of force as an on-line TCM. Research has been done on feed- and cutting force, and dynamic and static forces. Force has been used to detect tool wear and tool breakage, and has been implemented in milling, grinding, turning, etc. [73, 31].

To analyze the resulting force signal statistical methods has been used, and the signal has been transferred to frequency domain by Fast Fourier Transform (FFT) [31].

Force sensing is one of the fundamental indirect measurements in TCM, and will be in the future.

Vibration sensors

Vibration sensors (accelerometers) are another common sensor in TCM [73, 31]. The reason for this is that accelerometers are cheap and reliable. Unfortunately there has not been found any significant uses for accelerometers, one reason is that there will be vibration when machining, and these vibrations might interfere with the vibrations generated by changes in the tool due to wear.

The two most common analyses of the measured vibrations is to calculate the Root Mean Square (RMS) value of the signal in time domain, and the other is to transfer the signal into frequency domain.

The RMS value is a simple measure, when vibration increases, the RMS value will increase. This has been shown to correlate with increased tool wear, and a tool breakage will also create a pike in the RMS value [73, 31].

By transforming vibrations into frequency domain, it can be shown that vibrations at different frequencies behave differently as the tool wears [31].

Acoustic emission

During metal cutting the workpiece will undergo considerable plastic deformation. This will rip bonds between atoms apart, which will release strain waves. Acoustic emission (AE) is the collective name used for the energies released because of the micro changes in the workpiece [73, 31].

Because AE is a high frequency signal, from 150 kHz and upwards, the most common feature extracted from AE is the RMS value. A full featured sample of an AE signal will result in too much data if the signal is to be monitored on-line.

The most common research area for AE is to detect tool breakage [73, 31], but some have also done research on use of AE for detecting tool wear [31].

Optical sensors

The two most common optical sensors used are laser [73, 31] and vision [31]. Optical sensors are not applicable for on-line measurement, although some has done research on on-line laser sensing [66].

A commercial system today uses laser to detect missing and broken tools [24]. These systems are however not accurate enough to detect tool wear.

A lot of research on using laser and vision for tool wear detection has been conducted [31], though most results show high inaccuracies and are therefore not usable.

2.3.4 Research

Research on the properties of tool wear, and how to detect it has been an ever ongoing task. An early review [54] identifies the major sensors for tool state detection which are still used and done research on today. But there is still no intelligent system which can positively identify a worn tool under all circumstances.

Artificial Intelligence (AI)

One main area for research in the later years is AI. In metal cutting there are a lot of parameters which will influence the applied sensors. AI methods can be used to find unknown relations between various parameters and sensors.

The most common method is Artificial Neural Networks (ANN) [1, 22, 35]. ANNs can be trained by using well known good and bad values, and then the ANN can be used to classify further measurements.

Another commonly used method is fuzzy logic [32, 33, 22, 35]. There is often a set of small changes which indicate that the tool state has changed, and there is no direct mathematical property which connects these. By using fuzzy logic a connection between a set of properties can be made.

Signal filtering

Another area for research is signal filtering [19, 67, 75]. Signals measured from in-process sensors often contain a lot of noise which is not correlated to the tool state. By filtering out the noise a better signal can be provided. This, together with AI methods, will improve TCM.

Vision technology

Modern digital cameras have made the quality of pictures much better. With increased computation speed and research on filtering technologies and algorithms [42, 70, 76], there has been progress in identifying tool wear by vision. This technology has not been perfected yet, but is promising.

2.3.5 Industrial Applications

Even though the ultimate TCM system has not yet been developed, there are commercial systems which are in use today [21, 25, 24]. The most common use is in high volume processes which run unmanned for long periods. A worn or broken tool in such systems will result in hours of wasted work and materials, and might result in broken machinery. Therefore the investment in time and resources to find the correct parameters and sensors for a TCM system is worth it [52].

Part I

Logging of Events in Automated Production

Chapter 3

Infrastructure of Automated Production

3.1 Background

To create a modern production plant can be quite challenging. A complete product requires many different processes to be finished, and many different solution providers will supply machinery to do each process. Since the product must go through the whole production facility without being touched by human hands, it is necessary with conveyors and buffers to connect the different parts of the production. The amount of buffers and conveyors though should be kept at a minimum, since they do not add value to the product.

In order to control the complexity of such a mixed production environment it is necessary with a lot of computation power to handle each process, and it is necessary that the different parts of the production can communicate with each other, with the operators, and with the main plant control system.

3.2 Logical controllers

The backbone in a modern plant is the Programmable Logic Controller (PLC). The PLC was invented in the 70s, at the same time as the minicomputer. The common method of logical control in a production plant was large relay panels that was complex to rewire when the logic had to be changed. So General Motors asked several different suppliers to create a solution that easily could be programmed.

The PLC has a very simple main functionality. Read input values from different sensors, execute the main program, and then set output values to different actuators. Do this as fast as possible, which should be in the range of 20-100 times in a second. The input values are often either a current $4 - 20mA$ or a voltage $0 - 5V$. During the PLCs 25 year life the main functionality of the PLC has not changed, but the PLC has become a very complex product. The main PLC vendors deliver a large array of different products suited for different tasks and the PLCs can communicate with a lot of different external equipment.

A simple PLC program might be to fill a box with products from a conveyor. When a product is sensed on the conveyor, stop the conveyor and actuate the picking manipulator. When the manipulator has picked up the product, start the conveyor again. After the manipulator has delivered the product in the box, check if the box is full. If the box is full request a new box. In addition we need sensors to check if a box is ready.

PLCs are sold as stand alone controllers, but they are also integrated into most of todays production machinery. Hydraulic presses, milling and turning centres, and most robot controllers. So the number of different controllers will be quite large. So it is necessary with main PLCs to control the different machines and a network which can be used for information exchange.

3.3 Industrial networks

The first PLCs read their input and manipulated their output by analogue signal cables, often referred to as 4-20 mA current loops. With these analogue signal cables a current of 4 mA represented a low signal, whilst 20 mA represented a high signal. As an example the signal could represent the speed of a motor: 4 mA would be transmitted if the motor was standing still, 12 mA when running at half speed, and 20 mA when running at full speed. If a PLC is connected to a lot of different input sensors and output actuators, a lot of cabling is needed. The solution to this problem is to use a more sophisticated network where several sensors and actuators can be connected to the same cable.

In addition it might be necessary for PLCs to communicate with other PLCs and more complex devices like robots and CNC machines. The different kind of networks can be placed in four basic categories [50, page 22]:

- Sensor Networks, protocols designed to support discrete I/Os
- Device Networks, protocols designed for process instrumentation
- Control Networks, protocols designed to connect controllers and I/O systems
- Enterprise Networks, protocols designed with focus on IT applications

The first industrial networks emerged in the late 70s, designed to serve the first two purposes. Each PLC vendor created their own fieldbus standard, which quickly became a problem since it is necessary for PLC from different vendors to communicate. During the 80s a consent on some fieldbus standards emerged, so today this is not a big problem. But it is still an issue to look out for when procuring production machinery.

> Fieldbus is a digital bi-directional, serial-bus communications network supposed to link various instruments, transducers, controllers, final control elements, process stream analysers, SCADA, and computer control systems. [50, page 11]

The main characteristics of a fieldbus is determinism, high data efficiency when transferring small amounts of data, and low latency. Some fieldbuses specify the protocol and the transfer media, like CANbus and Profibus, whilst others only specify the protocol like MODbus and DeviceNet.

Profibus-DP (the most common Profibus variant) is a master-slave network. In each main PLC cycle the master sends data to each slave, and will request the slave to return data. All I/O is setup in front, so the bounds on the data transmission is known. In a Profibus network there might be 127 nodes, and the packet size is 256 bytes with a 12 byte header. The main setup is with one master, but it is possible to switch masters by token passing.

With all these different interfaces a modern industrial network often has a pyramid shape, with sensors and actuators (data producers) at the lower level and databases and applications (data consumers) at the highest level. Between these two levels there might be several intermediary ones, interconnected by gateways and bridges. Gateways and bridges make data intercommunication difficult, and in order to improve integration between levels companies are encouraged to use compatible networks.

3.4 Human Machine Interface

PLCs and other advanced production controllers are small black boxes with some small LEDs to show operation status. In order to make it possible for operators without deep knowledge about programming to understand what is going on, complex Human Machine Interfaces (HMIs) have been developed.

Today the HMI is a computer with a visual representation of the production cell on screen. In a big cell the representation might be coarse, with the possibility to dig down into each part for a more detailed view. The visual representation is often color-coded to show the parts of the production cell which is working, and the parts need intervention. An intervention might be to refill a feeder, or fix a machine that has stopped. Many HMIs also show some statistics about the current production, and have the possibility for some simple interaction with the production cell. Start/stop, select current product etc.

3.5 Supervisory Control and Data Acquisition

In addition to the HMIs which show a real-time view of the production equipment, and are mainly a help for the operators, more complex logging applications have also been created. Supervisory Control and Data Acquisition (SCADA) is a higher level application which is designed to gather information about the production and store it for later analysis.

Information like the temperature in an oven, the number of parts produced, how many parts are produced with flaws and which flaws etc. might be interesting data.

SCADAs and HMIs have become more similar during the years. Computers have become more and more powerful, and they can accomplish everything needed for both HMIs and SCADAs. So a HMI is often a stripped down SCADA with a smaller price tag.

The main work for the SCADA system is to gather data which indicate when the production facility is not working as it should. SCADAs are also often created by others than the PLC and fieldbus providers, so they are designed to work in a mixed environment.

3.6 OLE for Process Control

As mentioned earlier, there are many different vendors for PLCs, industrial networks, SCADA systems and HMIs which need to communicate with each other, in addition to the need of developing custom applications that communicate with industrial equipment. As there are numerous vendors of automation equipment which all use their own protocols to communicate with, in addition to the large variety of network interfaces, it is difficult to write software which can be used for all variations. To make it easier for software developers to communicate with this large variety of automation equipment, a standard communication interface was needed. With a standard communication interface the software developer can create programs which are hardware independent. Independence is always good as it simplifies upgrades, as not all components have to be updated at once.

With a standard communication interface, this interface will need a driver for each different type of hardware. This driver can be written once and then reused by all software programmers. So a standard communication interface will also reduce the time and resources needed for creating drivers. Since they are written against a standard, they can be created independently from the software which will use the drivers.

To be able to create a standard and implement it quickly, the industry decided to use Microsoft's Component Object Model (COM) and Distributed COM (DCOM) as the basis for the communication interface. The communication interface was named OLE for Process Control (OPC). COM is used to communicate between processes on one computer, whilst DCOM is used to communicate between processes on different computers. So OPC works as a server/client solution, where the user creates clients which ask for data from the server. The server can be on the same computer as the client (COM), or on another computer (DCOM). The server then handles all communication with the automation equipment. The main benefit of the server/client model is that one server can support several clients.

Today OPC is often the interface of choice for HMI and SCADA systems.

3.6.1 OPC interface details

The are also some disadvantages with a standard interface. The interface might be too simple, then it wont satisfy the varying needs of the users. The interface might be too complex, then it will be more work creating drivers, just to implement functionality which most users do not use.

To simplify the implementation of OPC servers and drivers, the OPC specification is split up into several different interfaces. The most common are Data Access (OPC-DA) and Alarms and Events (OPC-AE). OPC-DA defines interfaces to read and write data synchronously and asynchronously, and to listen for data change events. When listening for data change events, the user registers which variables he is interested in at an OPC-server. A update rate is agreed upon between the PLC, the OPC-server and the user. The OPC-server then polls the PLC at the agreed update rate, and when the value changes it is reported to the user with a timestamp. The timestamp is created by the OPC-server. If an analogue value is monitored it is possible to set a deadband for which changes are not reported. This is to avoid that every small changes propagate to the OPC-client.

3.6.2 Shortcomings of OPC

COM is an open standard, but not many software platforms have implemented this technology. So today is using COM the same as using Microsoft Windows. As Windows has had a (deserved?) reputation of instability, Windows has never gained ground as a controller in automation software. So OPC has mostly been used to collect data for storage and for display. This might have been an hindrance for increasing the interest in data collection, as the implementers of automation functionality often have limited knowledge about Windows, and therefor often have little knowledge about OPC and the possibilities of data collection.

There is currently work on a new revision of OPC which will remove COM as the underpinnings, and maybe the new revision of OPC will be platform independent.

Another issue with OPC and the Windows requirement is that there is no Real-Time guarantees in Windows. This means that there is no guarantee that Windows will connect to the PLC at the specified time interval, and there is no guarantee how long the time interval will be from the request is made and to the data are timestamped. The effect will be that even with a signal which changes at a constant rate, the resulting timestamps might vary. How much deviation there will be is dependent on the priority of the processes, and the load on the computer running Windows.

3.7 Overall Equipment Efficiency

In this chapter there is mentioned a lot of different equipment and technologies for managing a modern interconnected production environment. The main reason for creating all these technologies and making production equipment so hi-tech, is to improve efficiency. The product vendors all brag that with their technologies it is possible to create a production plant where the product is untouched by human hands. This will lead to a high quality product and the production will be very efficient.

This is of course not the whole truth. Hi-tech equipment is complicated, and with the evolvement of flexible manufacturing there are a lot of components which may fail. And even when they work it is not very easy to make everything interact well.

In section 1.4 Overall Equipment Efficiency (OEE) was introduced as a much used benchmark for measuring the efficiency of manufacturing equipment. This section will describe the theory behind OEE and how it can be measured.

The basic formula used for calculating OEE is:

$$\text{availability} = \frac{\text{actual production time}}{\text{planned production time}}$$

$$\text{performance efficiency} = \frac{\text{produced amount} \times \text{theoretical cycle time}}{\text{operating time}}$$

$$\text{quality rate} = \frac{\text{produced amount} - \text{defect amount}}{\text{produced amount}}$$

$$\text{OEE} = \text{availability} \times \text{performance efficiency} \times \text{quality rate}$$

An OEE of 85% is considered world class. Two main areas to consider when improving OEE is to look at unexpected downtime and unexpected waiting time. Unexpected downtime is when the production facility does not produce because one or more machines are down and halting the entire production. Unexpected waiting time is when a machine is idle waiting for one or more components to be fed to it.

Even though OEE is a valuable tool to measure performance, it has some shortcomings. One is that there are no standard definition of the different elements of the OEE calculation. If the theoretical cycle time is set to the current cycle time of a system, and then the cycle time is improved, it is possible to get a performance efficiency higher than 100%. So OEE cannot be used directly as a measurement between two different factories, unless they use exactly the same method for calculating the OEE.

Another shortcoming is that OEE can only indicate where there are problems in a production line. The elements which perform badly will be highlighted, but OEE does not tell anything about what the problem is. So when an area of problem is identified, another round of analysis has to be done to find out what the cause of the low OEE is. This is important to keep in mind when implementing OEE analysis.

3.7.1 Systems for OEE

Systems for calculating the OEE exist, and are used in production today. They are mostly implemented as an addition to already existing SCADA systems. Some details like planned production time and theoretical cycle time have to be entered into the system, but the rest of the data is collected from the production equipment.

The missing data are then how many parts are produced and if they are produced with or without errors, and for which time periods the production lines have been activated and for which time periods there was an unscheduled downtime. None of these data require a high resolution, a detailed timestamp or a large data storage. Since most events entering a SCADA system are timestamped (See section 3.6 about OPC), the necessary data are available either directly or by a small program in the SCADA system.

The extra features, often implemented by OEE software to add value in respect to standard SCADA systems, are visual tools that describe each component of the OEE calculations: Availability, Performance efficiency or Quality rate. By highlighting problem areas it is easier to identify where resources are to be set in to improve the situation.

Chapter 4

Data Collection

In this chapter the need for computation power, network interconnection and data exchange to fulfill the needs in a modern production plant will be discussed.

4.1 Background

In the previous chapter different data consumers and methods for industrial data exchange were discussed. Most of the technologies and products are based on ideas from the early days of PLCs and industrial networks, and they date from the late 80s or early 90s.

Since then the development in computer power and network technology has leaped forward. The speed of CPUs double every 18 months, so a CPU is 20 times more powerful today then 15 years ago. The main network technology in the IT industry, Ethernet, has increased its speed from 10 Megabit to 10 Gigabit and has become much more reliable.

With this increase in computer power it is possible to do things not imaginable years ago. With the increased demand for not just automating production, but also to make the automated production cells work at 100% efficiency, a much better knowledge about production equipment is needed.

It is therefor necessary to collect data from production equipment with a higher rate and a higher level of detail than ever before.

New data consumers are for example simulation models etc.

4.2 Data storage

The most important thing when collecting data is to store them for later use. If detailed data for a large production cell is to be collected and stored, it will very quickly become a very large amount of data.

If we have a simple assembly where we want to know the state at any time we need 1 byte to store the state (with a maximum of 256 states), 8 bytes to store a timestamp with millisecond accuracy (8 bytes is the standard in many different computer systems), and 4 bytes for a unique production cell identifier (enough for 4 billion numbers, $0 - 2^{32-1}$). If the production cell has a cycle time of 2 seconds, and it goes through 4 states, we will produce $(4+8+1) bytes/state \times 4 states = 52 bytes$ every 2 seconds. For a year with 24/7 production it will be $52 bytes \times 30 cycles/minute \times 60 minutes/hour \times 24 hours/day \times 365 days/year = 782 \times 2^20 bytes \approx 1 Gigabyte$. In addition there will be some overhead by the storage engine and some metadata to describe what is stored.

With a dataset as described above the complete information about what a production system does should be available. Often it will be enough with a smaller dataset, but with a complete dataset it is possible to calculate precisely how much time the production system has spent in different states, and where it is possible to improve the system. After the production system has been used for a while and the performance indicators which describe the system status are known, they can be calculated on a regular basis thereby shrinking the size of the dataset.

In a large production facility there is not only one production cell, but many cells with many systems. And each system can have several states and also produce other interesting information like the results of different operations and the results of different measurements etc.

A modern disk can be as large as 500 Gigabyte, and simple RAID controllers can join 4-8 disks. If needed more expensive solutions with even larger capacity can be procured. These large disks make it possible to store the vast amount of data required to describe the events and results in production cell in detail.

4.3 Network technology

As mentioned earlier Ethernet is the network of choice in the IT industry. This has made Ethernet network solutions cheap and fast. The media industry like video and television wants to deliver high resolution pictures to the consumer whenever and wherever the consumer wants. Movies must be shown in real-time and they can use unlimited bandwidth, so a fast reliable network is required.

In its infancy Ethernet was quite fast (10 Megabit pr second), but very unreliable. In an Ethernet network there can be an unlimited amount of nodes, and there is no one to control which node is allowed to send data. This results in data collisions in the network and packet loss.

Today the most common network topology used with Ethernet is the star. Many different computers connect through one main hub. This hub can again be connected to other hubs. The first hubs were not very sophisticated, they just forwarded incoming data from one incoming port to one of the other ports. Modern hubs, called switches, have intelligence. They know which computers are connected on the different ports, and they can handle much more data per second internally than the network speed. So if 4 computers are connected to a switch, computer 1 and 2 can communicate without disturbance at the same time as computer 3 and 4 can communicate.

So modern switches have made the in theory very unreliable Ethernet network to a very reliable and very fast network. 1 Gigabit Ethernet is common today, and 10 Gigabit will be available in a few years. A reliable ethernet based network does not drop packets, and by calculating the maximum number of nodes in a network and their maximum transmission it is also possible to find an estimate of the maximum transmission time.

These extreme speeds and the low cost of Ethernet makes Ethernet very attractive for industrial networks. Most PLCs and computers are already equipped with an Ethernet port to connect HMIs and SCADAs, but it is becoming more common to use Ethernet as the high-speed reliable connection between different PLCs as well.

4.3.1 Protocols

The backbone protocol used with Ethernet is the Internet Protocol (IP). The IP header describes the contents of the data enclosed in this package, and source and destination of the package. Other base protocols are the Address Resolution Protocol (ARP) and the Internet Control Message Protocol (ICMP). These protocols are used to maintain the network structure in a network.

On top of IP there are two common protocols, User Datagram Protocol (UDP/IP) and the Transfer Control Protocol (TCP/IP). UDP is a very simple protocol which wrap data with a source and destination port, and the size of the data before sending. UDP does not guarantee any delivery, so if a packet is lost, the programs using UDP must have a fallback to recognise this.

TCP/IP is the most common protocol used with Ethernet. TCP creates a connection between a source and destination, maintains this connection and guarantees that packets will arrive at the receiving program in the order sent. In earlier Ethernet networks with a high packet loss this protocol could require much resources, but in a modern switched Ethernet network it will be reliable and fast. The only thing which is slow is creating a connection and tearing it down. So it is important to initiate one connection and reusing this in situations where high speed and low latency is required.

4.3.2 Transmission efficiency

An unsatisfying side with Ethernet and TCP/IP is the data efficiency. The size of a package can vary between 64 and 1500 bytes when using TCP/IP. Ethernet itself uses bits for data maintenance, IP uses 20 bytes for the headers, and TCP uses another 20 bytes for its header. Profibus uses only 12 bytes for headers, but with a maximum packet size of 244 bytes. So for small amounts of data TCP/IP has a large overhead, and is inefficient when compared to Profibus. Luckily we need larger data sizes, and the speed of Ethernet (1 Gbps vs. 12 Mbps) removes the problems associated with the large overhead.

4.4 Real-Time

The notion of time is fundamental to our existence, although we do not have any sense of time. We always live in the present, but time makes it possible to talk about, and order, events in the domain of time. For computers time is very hard to comprehend, as computers are event driven. Everything is done in sequence without relation to time.

Most production systems work in real time, as we humans understand time. So it is very important that computer systems used to control real time systems understands time. If there is a stimulus from the environment on the production system, the computer system has to react on the event and return a correct result within a certain time limit. If the time limit is exceeded, the real time system has failed.

In addition to being real time, many production systems also work as distributed systems. In a distributed system there are several nodes which all can report about events. In such a system it is immensely important to have an understanding about time, so that events can be handled in correct order. If events are processed in wrong order, the control system will end up in a state which does not respond to the real world.

As mentioned, a real time control system has to respond within a certain time limit to be able to function correctly. The ultimate solution would be if a control system would be able to respond "NOW". This is unfortunately not possible. Even the human control system (the brain) cannot work that fast, even though it can react to very complex problems. If we put our hand on a hot plate, our reaction is not fast enough to prevent us from being burned. One incredibly fast response system is the one used in airbags. After the sensor detects a sudden retardation (crash), the airbag inflates in 0.015 seconds to the size of a beach ball.

An example which shows that "Real-Time" does not have to be measured in milliseconds might be a system to prevent an invaluable vase from hitting the ground. If the vase is standing on a 1 meter high table, how fast must a security system react if the vase has to be caught before it hits the floor? From dynamics we know that $s = 1/2 \times a \times t^2$, where s is distance, a is acceleration and t is time. The time is then $t = \sqrt{\frac{2 \times s}{a}}$. If s is 1 and a is about 10, t is then $\sqrt{\frac{2 \times 1}{10}}$ equals 0.45 seconds. Response time faster than this is not necessary.

$$\text{Real-Time Systems}\begin{cases}\text{Soft-RT Systems}\begin{cases}\text{High Availability}\\ \text{High Integrity}\end{cases}\\ \text{Hard-RT Systems}\begin{cases}\text{Fail Safe}\\ \text{Fail Operational}\end{cases}\end{cases}$$

Figure 4.1: Classification of Real-Time Systems

Figure 4.1, from "Distributed Systems" [55], shows how real time systems can be classified. Most functions of a production system must be Hard-RT and fail safe. The control system in robots and manipulators has to be Hard-RT to make sure that trajectories are followed, and that end points of movements are correct. If a manipulator moves past an end point, the result can be a catastrophic crash. Other main events like pallets triggering presence sensors have to be caught, to make sure that pallets are stopped and released as they should, etc.

The main loop of the control system which decides what to do next does not have to be very quick, but time lost while the control system decides what to do next is time lost for doing productive work.

So a production system is based on small building blocks which need to react in real time, with a control system above this which need to work "real fast". So when collecting data from a real time system using a non real time system it is important to handle the flow of information without disturbing the critical tasks in the real time system, and without causing too long delays in the main control logic.

The best design for interaction between a real time and a non real time system is that the real time system keeps track of events, with their time and ordering, and then push the data to the non real time system when idle. One solution to this problem is to make the real time system multi tasking with the possibility to preempt tasks. Then the movement of data can be done in a low priority task, which can be preempted when the main task needs it.

Preemption means that it is possible to switch a running task with another and then restore it later. Multitasking and preemption are very nice principles, but they create overhead in the control system. The control system will need more memory, a better CPU and the minimum response latency will be higher than in a system doing only one task.

4.5 How to collect data

Earlier OPC was discussed as the tool of choice when it comes to collecting data from a production environment. Yet OPC does not do well enough when it comes to high-speed detailed data collection.

OPC was designed in an era where master/slave networks were standard, the amount of data collected was much smaller than today, and PLCs were much slower. OPC has two main problems because of this: OPC polls units for information, and OPC timestamps events at the computer. This setup works fine in a master/slave network where we do not want to stress the PLC, and the timestamp for the event does not have to be very precise.

This is not good enough anymore. When polling for values we might lose changes. If a value is checked every 50 milliseconds the value might have changed many times in those 50 milliseconds, and it can be the same as it was the last time it was checked. So we lose much information. If the value is timestamped at the OPC server we add uncertainty about when the event actually happened, and there must be constant time from when the event happens to the time when the OPC-server gets the event for the timestamp to have any value at all. The uncertainty for the timestamp will come from three sources: The PLC can delay the information transfer, the network can have a variable transfer time, and the OPC-server is in the hands of the operating system scheduler.

Ethernet is often referred to as a producer/consumer network. This means that all nodes connected to the network can transmit at any time. So if we use Ethernet the best way to transfer information from a node to a data collection server is to initiate the transfer from the node.

A theoretical setup of a modern and efficient data collection system would be something like this: The server creates a connection to a node, and requests that some information should be sent when a change occurs. The node is hard at work doing its main control cycles and communicating with other nodes to trigger the actual production. When an event occurs that should be logged, the identifier of the event, the timestamp of the event and the value is registered for transfer to the logging server. The data is sent when the node has some idle time. The point of time when the actual transfer occurs is not very important, because the timestamp of the event is already stored.

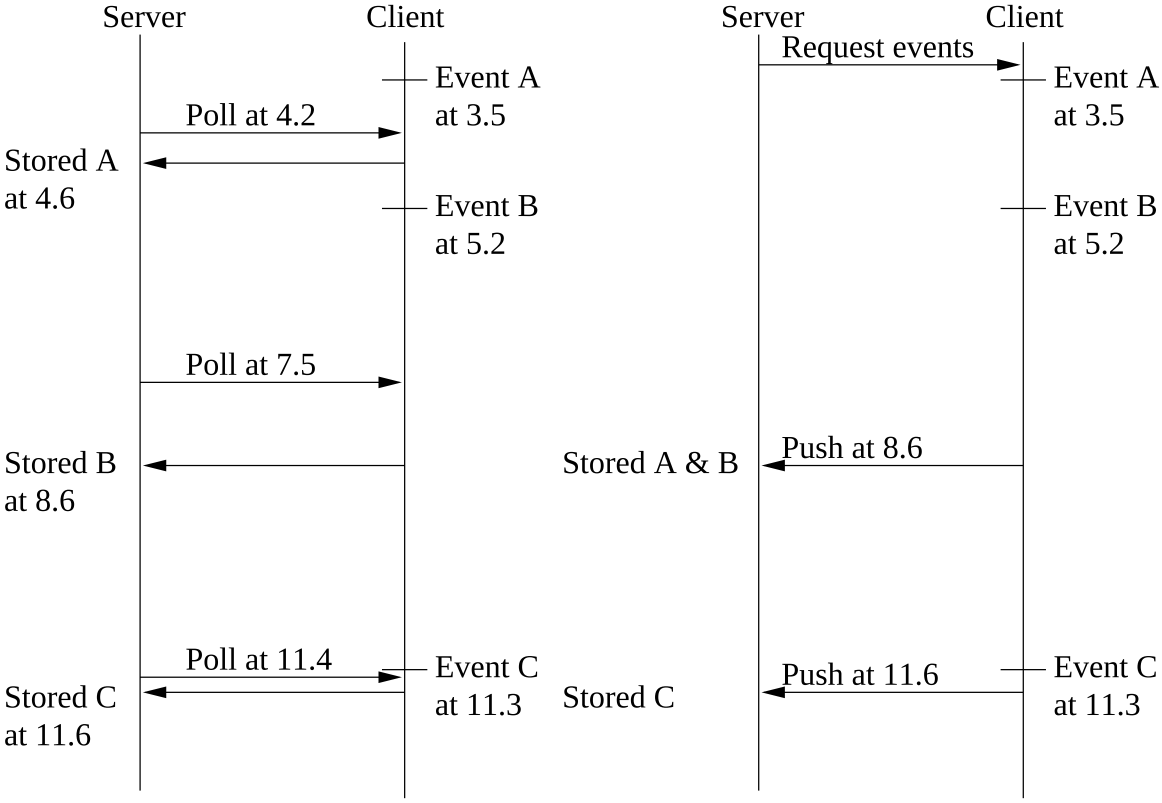


Figure 4.2: Poll and push data collection

4.5.1 Reliable timestamps

When using OPC we get comparable timestamps. Information is gathered as quickly as possible, and timestamped by one source.

If the timestamps are created by different nodes in the network, they could all have varying clocks. So it is important that all the different nodes synchronise their clocks. Clock synchronisation is an old and well debated problem when it comes to distributed networking. At present one common standard has emerged for synchronising clocks in a TCP/IP network, the Network Time Protocol (NTP, [59]), which is described in "Distributed Systems: Concepts and Design" [30, pages 389–392]. This protocol is used to synchronise clocks on the internet, where there exists a set of servers dedicated to be the correct clock source. In a local network where the user has control of all the nodes, it is possible to get an accuracy of 200 microseconds. Most PLCs have a main cycle time about 10 milliseconds, so the accuracy of NTP will only be a fraction of this.

4.5.2 Event ordering

Another interesting problem concerns the order in which events happen.

An ideal logging system should transfer data asynchronously from the nodes to the logging server. This means that there is no connection between when an event happens and when it arrives at the server. So we lose the event ordering. Since the network is not 100% reliable and there might be some delays, the order in which data arrives at the logging server is not guaranteed to be the order in which events happen.

If events can happen so close in time that the timestamp is not good enough to split them apart, logical clocks can be used. A logical clock is a counter which is incremented each time an event occurs in a system. If this counter is synchronised between each node, each event will get its position in the event order.

There is a trap, concerning the synchronisation of the counter. The synchronisation also uses the network, and this requires time and is dependent on the network which still is not guaranteed. The logical clock problem is another well debated problem, and several solutions have been proposed.

Consistency in a distributed system is further discussed in "Distributed Systems" [55, chapter 4].

Chapter 5

Discrete event logging

5.1 Introduction

Production lines in the manufacturing industry can be represented as discrete event systems. In a discrete event system the operation is represented by a set of events which transform the system from one state to another.

Basic states often used in manufacturing systems are processing, waiting and downtime. These states can again be split into several others. A process can wait for upstream parts to enter, or it can wait for downstream resources to accept input. Or it can wait for an operator to do some work, or wait for a forklift to pick up some goods. The production process in itself will always consist of several states. A pick-and-place process typically consist of a movement to the pick position, a gripper close, a movement to the place position, and a gripper open.

The complete set of states in a production line, with several different processes and several parts entering and exiting, will quickly become very complex. It is therefor important to make system borders. Inside the system border the process is dissected into a large set of states, but to the outside it will only present a few basic states. Then the complete system will only interact with system borders, and inside a system border an independent system can be created.

5.2 Simulation

A discrete event description is often the first that needs to be done when designing a new production line. This description is important to get a good understanding of what elements the production line will consist of, and how they will interact.

Since a discrete event system will quickly become very comprehensive, it is difficult to understand all connections, and how one event will affect the complete system.

Discrete event simulation is a way to do testing on a discrete event system before it is realised. A discrete event simulation will force a complete review of all critical components in a production line. As production lines quickly become quite complex this is a valuable tool to test if all aspects has been thought trough, and whether the planned production line will produce the planned output.

But since it is impossible to know all parameters and data needed for a simulation model, discrete event simulation models are often superficial. If a model created during the design of a production line is to be reused after construction to test the effects of various changes, it will often be to inaccurate to give a precise answer.

5.3 Production lines

After an initial design and simulation of a production line, it will be manufactured and programmed. As earlier mentioned, PLCs are the standard control unit in modern production lines. A PLC is usually programmed as a state machine, so with a clever design, it is possible to match the states from the PLCs to the states used in the discrete event system description.

This connection between discrete event simulation and actual production lines is something that is in the early stages of research and development of simulation systems. In finite element simulation a lot of experimental knowledge has been fed back into the simulation programs, but in discrete event simulation this experimental base is often missing.

If states in a simulation model is matched with states in a production line it will be possible to feed back real data from the production line into the model. With this feedback the simulation model will represent the actual production line, and it is possible to get good results from the model if some parameters are changed.

With increased knowledge about which states a production line is in, it is possible to detect which sequences are to slow, and if the system uses to much time in non-productive states.

To create a system for optimisation of production lines, and to create a better firmament for simulation, two experiments were conducted as a part of a large research project. The first experiment was conducted in a test lab, whilst the second experiment was done on a prototype production line which was used at the company for whom the research was conducted.

The first experiment was done during a research project funded by the Research Council of Norway, with NTNU, SINTEF and RTIM as participants, for Raufoss United. The main goal of the project was to industrialise new couplings, based on state of the art manufacturing and assembly technology. The second experiment was done at the prototype cell at Raufoss United, with focus on finding solutions which could be implemented on the final line.

5.4 First Experiment

5.4.1 Introduction

Raufoss United produces couplings for air brake systems and they are introducing a new generation of couplings based on composites. The research project is a large project which has lasted for some years, and will still last for some years to come. The goal of the project was to design highly flexible production and assembly methods with short changeover times. A key element has been assembly, but it is important to include design and part production in the process to design a complete production system.

The first part of the project regarded design of the new products with focus on design for manufacturing. The second part of the project regarded general ideas about plant layout and production philosophy, like tight or loose coupling between different parts of the production: composite moulding, brass machining, and assembly.

The second part of the project focused on general principles for assembly. Traditionally high speed pick-and-place assembly operations have been solved with special purpose machines designed to produce a small variety of products, thereby limiting the reuse of the equipment and enforcing high amortization rates, as discussed by K. Feldmann and S. Slama [34]. The reason for choosing these types of assembly solutions is their speed and accuracy, and relatively low cost compared to more flexible assembly systems.

For this project flexibility was a key issue. The new product program is assumed to consist of 50 to 100 different products. Each product will consist of at least 5 components, and it is estimated that the program will consist of 200–300 different components. The full volume production is expected to be 30,000,000 products per year, with a volume range from a few thousand to several million products per year.

By utilising modern industrial robots and sensing technology it should be possible to create a flexible system that can compete in speed and accuracy with even the most accurate mechanically controlled assembly machines. In addition industrial robots offer flexibility in both software and work envelop, which can be utilised to develop flexible and cost effective assembly systems even for high mix and low volume products. Lower efficiencies in flexible assembly systems are not caused by the robot itself but by the increase of non-productive time when switching between different assembly tasks.

If the entire life cycle of the product and its product family is considered, more expensive flexible assembly systems can be defended if the necessary product flexibility and ability to reconfiguration (reuse) of the assembly modules is secured. H. Makino and T. Arai [51] describe how a one cell flexible system, with high non-productive times, can be transformed, as need for capacity raises, into more specialised assembly lines with higher efficiency. However, flexibility can be kept by allowing reconfiguration of the cells in the line. This can be achieved by using standardised assembly cells with easy exchangeable product specific units.

One of the major challenges with flexible assembly systems is part feeding. Feeding can either be ordered or disordered. With ordered part feeding the cycle time will be stable, whilst disordered feeding might introduce variations. In "Computer Vision Strategies for Flexible Part Feeding" [57] the author discusses how vision recognition can be used for disordered feeding, and how this introduces variations in feeding cycle times. This problem can be counteracted by introducing buffers and pallet assembly. This means that variations in cycle time, and non-productive time like part transfer between stations and tool change, are distributed on the number of parts on the pallet. This effect means that the variation in cycle times introduced by flexible feeders, and the problem of balancing cycle time between stations, also is avoided.

To test the feasibility of using flexible solutions for manufacturing of couplings, a robot cell at NTNU was set up to do the complete assembly of one of the new couplings. By using vision and only allowing non-fixed resources, like the tools for the robot and the pallets, to be changed for new products, a highly flexible solution was tested. To find out if the solution was working well, and to reduce development time, a data collection system was created to gather performance statistics. The data collection system would also give detailed information about each part of the assembly cycle, so that the performance of a full assembly line with several robots could be predicted.

Two major challenges in this part of the project were the design of the tools and the data collection system. To reduce the amount of fixed assembly equipment needed, the tools had to pick parts and assemble them. In high speed assembly there is no time for reposition elements, and one design goal was to avoid use of any fixed resources for assembly.

The data collection system had to gather all interesting events, and it should be easy to extend the system with new events as the knowledge of the processes would change. Additionally all events should be stored with a very accurate timestamp. Accuracy for the timestamps is critical when each assembly operation lasts only a few seconds, and it is interesting to find and remove every wasted moment. Accuracy is also important when two solutions are to be tested against each other. Without accuracy it is not possible to do a good statistical comparison.

A criterion for a successful solution was to achieve an average cycle time of 2 seconds for each part of the assembly process.

5.4.2 Robot cell

Figure 5.1: Robot and the Flexfeeders

The robot cell consists of one Adept Scara S800, two Adept Flexfeeder 250 flexible feeders with stationary camera and one Flexlink XM pallet track. All equipment was connected to one Adept MV-10 controller with two mainboards. It is necessary with two main boards to get enough CPU power to control all the equipment.

The software on the controller is quite advanced. It is a multi-tasking OS with prioritised timeslices for controlling the movement of the robot(s). One controller can handle 15 axes shared on a maximum of 5 robots. Adept has since released an improved controller which can handle more axis.

The software on the controller is made of two layers. V+, version 14, is the operating system and the main programming language for the controller. On top of this layer is a programming environment called AIM, version 4.2C1. AIM consist of two parts. One part is a GUI where simple setup of a robot cell can be done, things like calibrating vision, defining feeders and pallets and defining locations and paths. In addition there is a set of higher level functions that make it easier to write programs for the robot. If the functions needed are not defined in AIM it is possible to write own functions in V+ and import them into AIM.

The major cycle in V+ is 16ms, which consists of 16 timeslices a 1 ms. The trajectory task has the highest priority in each timeslice, so that the robot movement will be smooth. Other main system tasks have a medium priority in some timeslices. The user can add some tasks, up to a total of 16 tasks (system and user), and assign priority for each task in each timeslice [12, Pages 56-66].

To program the robot a manual pad is provided, but the main programming interface is a windows-based program which communicates with the controller by Ethernet. The program is in real-time connection with the controller, and information like the current image and calculations from the vision system and the force measured is available in this program.

5.4.3 Assembly setup

The complete assembly (see figure 5.2) consists of one house, one cone to place inside the house, one locking ring and three o-rings. The three first elements are delivered on pallets, and the three o-rings are picked from the flex feeders. Since we only had two flex feeders, one of them had to deliver two different rings.

A total of 6 tools had to be used to do all the assembly. The tools where placed in a tool stand, and a standard tool changer was used to change between them. Since it takes time to change a tool, the assembly was conducted on pallets with 14 elements on each to improve performance.

Since this setup was not necessarily the same as the resulting setup at Raufoss United, it was important that each assembly operation could be analysed individually.

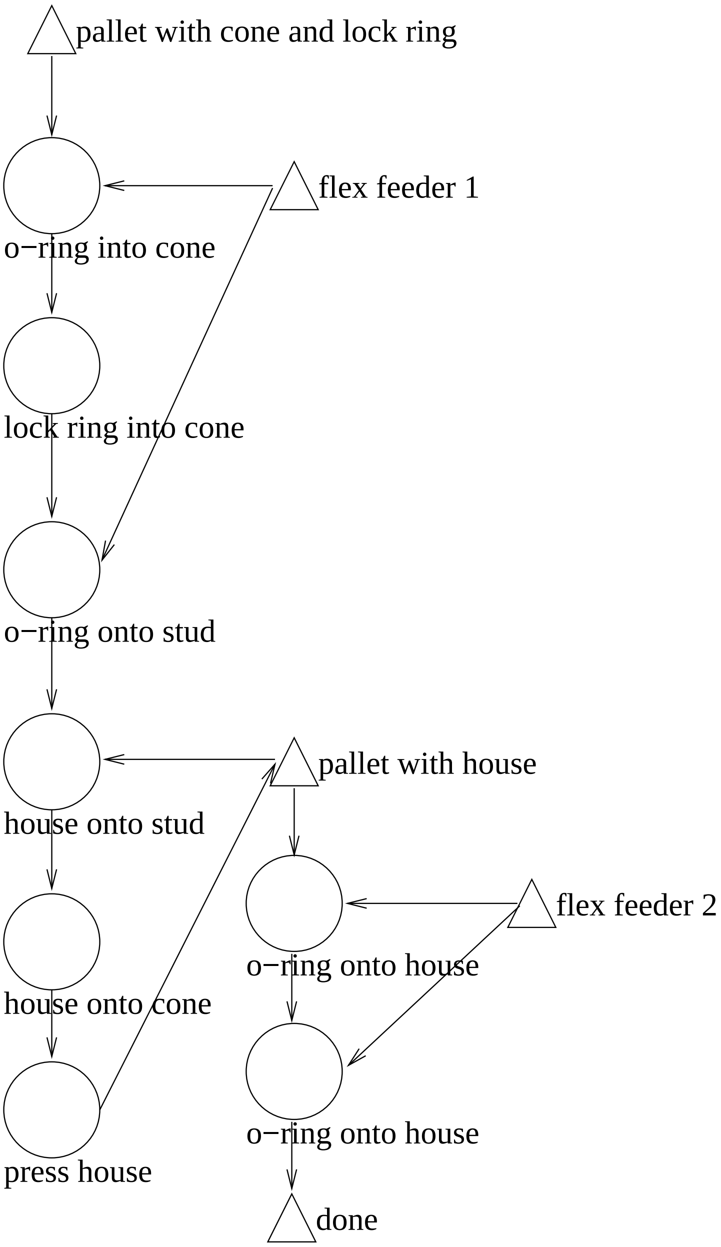


Figure 5.2: Process

5.4.4 Data collection

The interesting information was how long time each element of the complete assembly took, and when the robot did useful work and when it was idle. Several iterations of this were done, and the time for each element in the assembly was logged.

Data collection requires that data is created in the robot cell and that a computer collects the produced data.

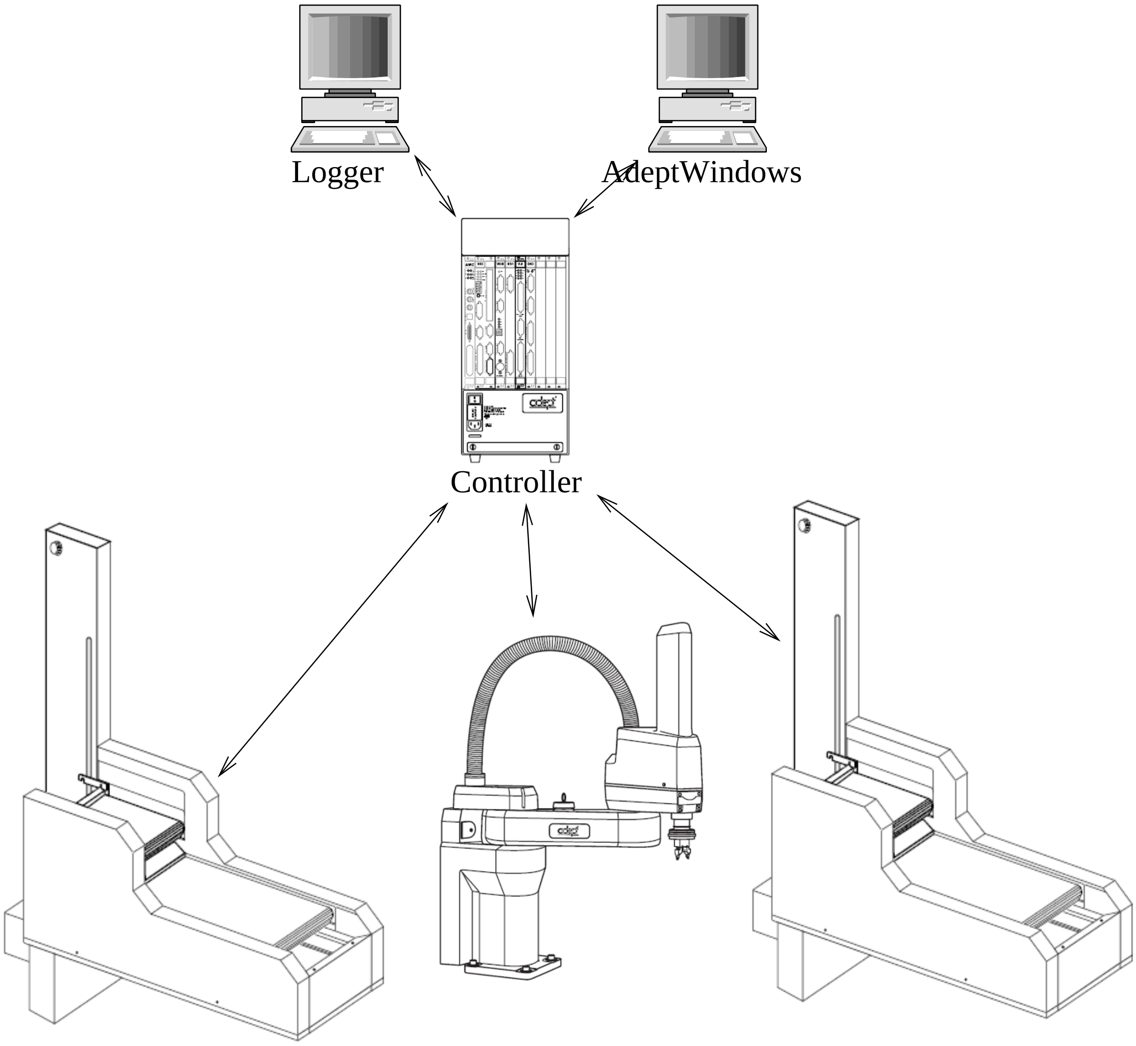


Figure 5.3: Computer interconnection

Figure 5.3 shows how the robot cell is interconnected with the computers. The connection between the computers and the controllers is Ethernet, whilst the connection between controller, robot and flexfeeders are designed by Adept.

Robot cell

To create the data needed in the robot controller was not too difficult. In AIM there already was a framework for creating events and logging them. The user must create a variable and register it for logging. When the robot program writes to the variable, the event will be registered by the logging system.

The logging system will log the name of the variable, the value of the variable, and the time of change. The information can be written to file, or to a TCP/IP port. The information is separated by comma and terminated by a newline character. So it is easy to create a daemon on a computer which listens to a TCP/IP port and processes the information.

To solve the logging problem one variable was created for each sequence part which was of interest to monitor. The variable was set high at the beginning of the sequence and low at the end. With such a consistent set up it was easy to manipulate the numbers afterwards.

The data transfer from the robot cell to the destination has high priority in only one of the system timeslices, so it will never interfere much with the main task of the robot cell which is to manipulate the robot. So the Adept controller is a good example on how data logging could be done.

Logging server

To increase the knowledge about the total process of collecting data, the logging server with software was designed and written and not purchased. This also made it possible to use standard low cost components and not industrial applications which usually have a very high price tag.

To make it simpler to modify the logging system based on the needs of the robot cell, the software on the logging server was based on open source software. Which, in addition to the advantages of making modifications easier, is free to use.

A standard computer was set up as a dedicated logging server. The computer had only a 233MHz processor and it performed well. So the hardware requirements are low, and a standard computer is sufficient. On this computer linux ([49]) and a MySQL ([56]) database was installed. A simple program was written in java ([40]) to process the information from the robot controller and insert the information into the MySQL database.

MySQL was chosen as the database engine since it is simple to install, and it is very fast for simple insertions. There was not any need for complex SQL to do the simple task of collecting the data from the controller.

The reason for choosing an SQL database at all is that it is possible to do many calculations in the database, and it is easy to connect other systems to it for information retrieval. A possible solution is to create a web interface, which was done. This is described in chapter 10.

As a result of the design and the components used, a low cost solution was made to collect the information. In the first phases of the design of a new cell a logging server like this will be a valuable tool. When the development and engineering is done, the simple logging server can be exchanged with a more complete solution, and the data can be imported from the simple system.

For inspection of the results, and to do calculations on the data, more tools are needed. The simplest solution is to export the data from the database into a spreadsheet.

Data entry

In the robot cell variables for different purposes could be registered. When registering a variable it was possible to decide that when the value of the variable changed, a message should be sent to the logging server. This made it possible to add variables which were triggered when interesting events happened in the assembly cycle. Like when the robot went into waiting position because there was no part available on the feeder.

Every event was sent as a tuple constructed of name, value, and time. This was stored directly into the database in one table. In addition the logging system has some understanding what the different variables meant so that it could keep track of some of the basic states of the robot cell, like when an assembly cycle began and when it ended.

This made it simpler to write queries which returned results as complete cycle time, average waiting time etc.

5.4.5 Results

Main cycle

The first tests conducted were to check how long each main cycle took. A timestamp was logged before and after the start of the main program, and before and after each main cycle. In addition the timestamp before and after each waiting cycle was logged. The waiting cycles occurred when there was no part ready to be accessed on one of the flex feeders.

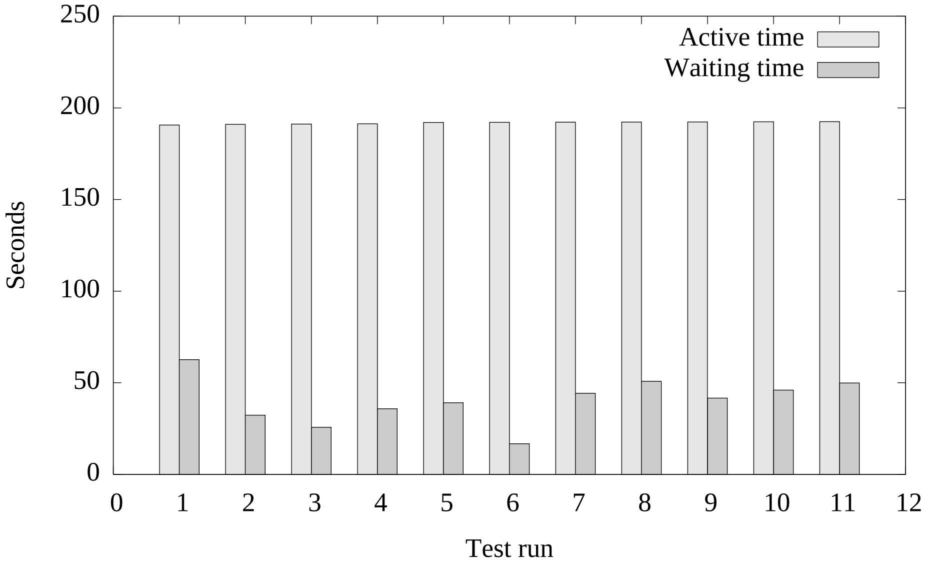


Figure 5.4: Complete cycle time

As seen in figure 5.4 the active cycle time of the cell is quite consistent, with an average of 191.80 seconds. The variation of 1.75 seconds may have many sources, but the main sources are probably a variation in image processing time for the flex feeders and the varying trajectories because of the parts being spread out on a $20cm \times 10cm$ large area on the flex feeders. In these test runs 14 elements were assembled, so the stability is very good.

The waiting cycle time in figure 5.4 has a much larger variation. The conclusion was drawn that the feed rate of the flex feeders are too slow. This could also be predicted by looking at the assembly. Later in the project there was done some work on fixing this, and the feed rate was improved considerably.

Main cycle parts

Figure 5.5 shows how the total cycle time is shared between the different elements of the total cycle. The main columns show the average cycle times, whilst the thin line goes between maximum and minimum time the operation took. There is also a tool change between each part which is not included in this plot.

Elements 2 and 4 of the assembly manipulates already positioned parts, so there should

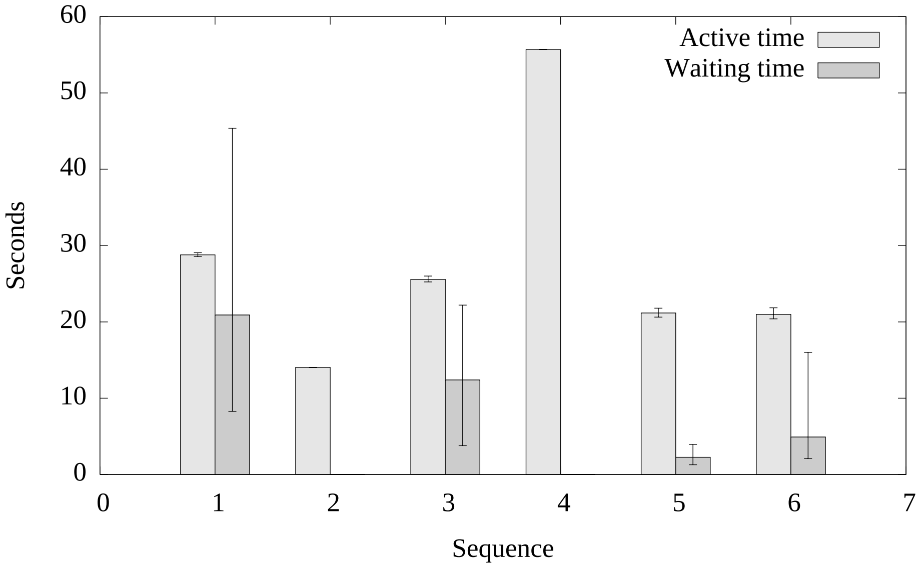


Figure 5.5: Average cycle time for the assembly parts

not be any waiting time. This is a good test on how well the data collection works. As seen in the graph there is no variation in how long the robot uses to complete these assembly steps. This indicates that the controller is not suffering from being a multitasking system, and that the timestamps are very accurate. To have an external check if the timestamps where correct, the whole assembly was recorded on video and the different time durations where calculated from the movie.

The assembly parts that have waiting time also have a variation in active time. This is because the trajectory varies. There are two sources for this, the first is that the pickup point for the o-ring varies, and the second is that the trajectory will vary depending on if there is an o-ring available. If the o-ring is not available the robot moves to a dedicated spot where it is not blocking the camera and where it is a short distance to the flexfeeder.

Detailed part

Introducing vision in an assembly operation introduces non-determinism. This was one of the main reasons for introducing pallets in the assembly. When assembling several parts in one cycle, the variations caused by the vision system will be dispersed, and there will hopefully be a stable average cycle time. But it is still interesting to find out how large the variations are, and if it is possible to reduce them. Even though variations cannot be avoided - and the assembly system should be designed with this in mind - large variations will disturb the optimal function of the line.

In addition the project had interest in testing assembly of one and one part. So a detailed study of the feasibility of using flexible feeders in single part assembly was interesting.

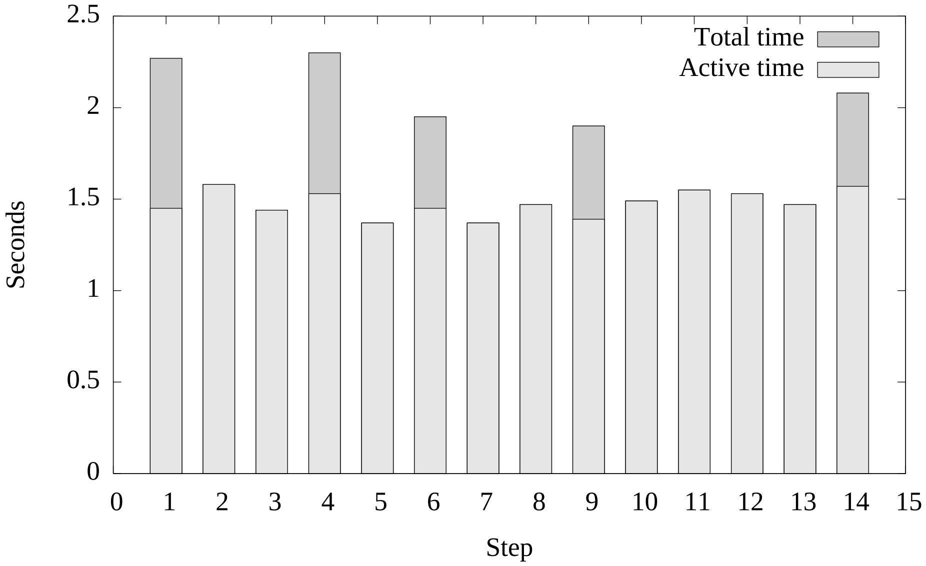


Figure 5.6: Detailed times for one cycle

To check if the flexfeeders could deliver parts at a high feed rate with low variation in time, signals were added during each assembly step to find the active and waiting time. Figure 5.6 shows a sample from one assembly. The robot program and setup was not changed, so this is the data collected when assembling one pallet with 14 products.

Other test runs gave better results, but this example shows how bad it can get with a non-optimal distribution of o-rings in the flexfeeder. The mean total time is 1.70 seconds, the minimum time is 1.37 seconds and the maximum time is 2.30 seconds. In a system where the target is a step time of 2 seconds this flexfeeder would cause large problems if it is used to deliver single o-rings, and not in groups.

The vision system can be operated in two modi. The first is to take one picture and then pick up all parts in the viewing area, the other is to take one picture for each part to pick up. Both modi have advantages and drawbacks.

If several parts are to be picked up from one picture it is very important not to move the other parts during a pick operation. When picking o-rings this is a large drawback since they usually lie close to each other, and the o-rings are expanded during picking. The advantage is that less time is used for vision analysis, and it is easier to plan when the belt should be indexed.

If a picture is taken for each part it is possible to pick all parts, but the disadvantage is increased time used for vision analysis, and it is difficult to pre-plan when to index. The system has to take one picture, analyse it to find no parts, and then order the index. During the experiments this was found to be the main source of waiting time in the assembly steps. The feeder was not able to index once and present a new part in the time the robot did the assembly of the previous part.

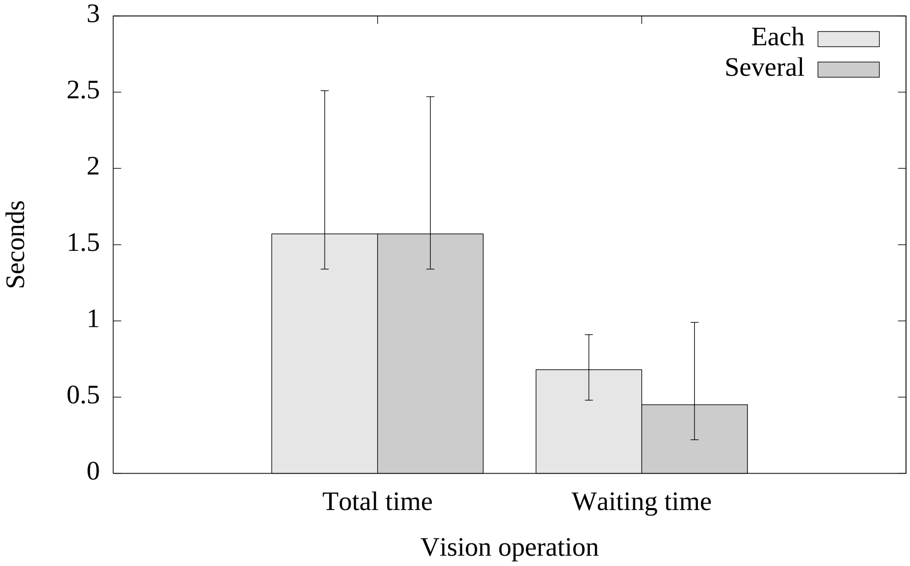


Figure 5.7: Average times for different modi

Figure 5.7 shows the average times with variation for the two different modi. To the left is the average total time used for the picking operation including waiting time. As can be seen the difference is minimal, so over time the different operation modi does not really differ. Even the maximum and minimum values are almost the same. To the right is the waiting time which occurred during the picking operation. This is the average waiting time of only the operations which actually had waiting time. So the total average waiting time is smaller.

The result is that when doing inspection for each ring, there will be less waiting periods, but the waiting periods will be longer. When doing one inspection for several rings there will be more waiting periods, but they tend to be shorter, although the single longest waiting period is longer when picking several rings in one picture than when picking one ring for each picture. The number of picking operations can be seen in table 5.1.

The main reason for the shorter waiting time for the case where several rings are picked for each picture is that the indexing can be preplanned. The main reason for the fewer indexing operations when picking one ring for each picture is that there is a lot of rings in the vision area which are situated very close to each other. It might be possible to add

	Total	With waiting	Percentage waiting
Each	392	28	7.14%
Several	196	49	25.00%

Table 5.1: Number of operations

.

a device which can increase the distance between the rings, but that will result in fewer rings.

And added benefit of picking one ring for each picture is that more rings are picked from the vision area, which again reduced the number of times one ring passes through the system until it is picked.

5.4.6 Impact on system

Data transfer in the controller is asynchronous. In a robot program a write to an IO-port does not have to be acknowledged, so the program can continue without interruption. So the only impact is from the timeslice reserved for TCP/IP communication. Since important processes have a higher priority than the TCP/IP communication, there should not be any large system impact by enabling data logging.

Unfortunately the code which does the logging of AIM variables is not accessible to users, so there is no way to check how it is actually done in the controller.

5.4.7 Conclusions

By using data collection a greater understanding of the system was accomplished. The data was collected by a computer using some relatively simple programs and a database.

In the experiments a timestamp was created each time the robot entered a new part of the main program and for each task in each cycle. By using logical controller terminology this can be described as states and state transitions. In addition the different tasks were identified as waiting time or active time. With this system the duration of each task in the robot program was measured. This information gave a clear picture of where the robot program could be improved.

Since many production processes can be described with states and state transitions, the approach described here can be implemented on a large variety of production systems.

In the next chapter the same system is used again on a different system without major changes.

To improve the complete cycle time, waiting time should be removed, and the duration of active tasks must be reduced to a minimum. During the experiments the logging system was actively used, and the effect of each change could be measured. The feeders were identified as the main problem regarding waiting time, and by improving these a remarkable speedup was achieved.

5.5 Second Experiment

5.5.1 Introduction

This second experiment was conducted on the prototype line at Raufoss United constructed after the previous experiment had been conducted.

The prototype machine was designed with one product on each pallet, so the main characteristics differed quite a lot from the experiments done previously with several parts on each pallet. With one part on each pallet the complete assembly cell has to be built with a dedicated assembly station for each assembly, because there will not be time to do tool changes.

In a cell with only one part on each pallet and many assembly stations tightly coupled it is very important that the time used for each assembly is stable and never exceeds the target cycle time of the complete cell. If one station uses too much time, this will affect all other stations as the pallet can not be moved. The only remedy is to have a buffer between the stations in hope that the slow station can do a fast assembly to keep up the cycle time.

When it comes to collecting data, it is very important to get accurate time measurements. The cycle time of the complete cell should be around 2 seconds, so each tenth of a second counts. After measuring the overall cycle times of each station, it will be important to measure the different sequences of each stations cycle to find out where it is possible to speed up the cycle.

5.5.2 Assembly cell

The prototype assembly cell is constructed around a very fast pallet conveyor. The conveyor has several subcells interconnected by short buffer zones. Each subcell has three assembly stations tightly connected. The index time in a subcell is 0.4 seconds, so it is important that the assembly stations can prepare their work during indexing.

Each assembly station has a specially designed manipulator and assembly equipment to do the assembly. For complex assemblies the assembly equipment is built in connection with the conveyor, and the manipulator just feeds the station with parts.

To control the assembly line and all the stations several standard PLCs are used. To control them there is one master PLC.

5.5.3 Data collection

Assembly cell

To collect data from the assembly cell, the master PLC was equipped with a Ethernet module. A PC was set up to communicate with the PLC and get the information required to analyse how long time different parts of the assembly took.

Unfortunately, the PLC did not have the same possibility of pushing events with timestamps as the Adept system had. The main option available was to use OPC. As mentioned earlier, OPC uses a polling strategy to collect data, so there is no guarantee to get all interesting events with the correct timestamp.

When programming PLCs it is usual to define states. Each part of the assembly is a state, and external or internal events can cause transition from one state to another. The state was stored in a variable, and this variable was monitored by the OPC server. It was highly probable that all state transitions would not be registered by the OPC server, but these would then be very short lasting and therefore not so important to register. Since the different state transitions are all known, the missing state transitions can be deducted from the state transitions which are registered.

Logging server

The logging server was again a standard PC. This time it was a newer and much faster computer. To collect the data a small program was written in C#, and the data was again stored in a MySQL database, see figure 5.8.

The OPC server was set up with a polling time of 50 milliseconds, so it should be possible to get a 100 millisecond accuracy.

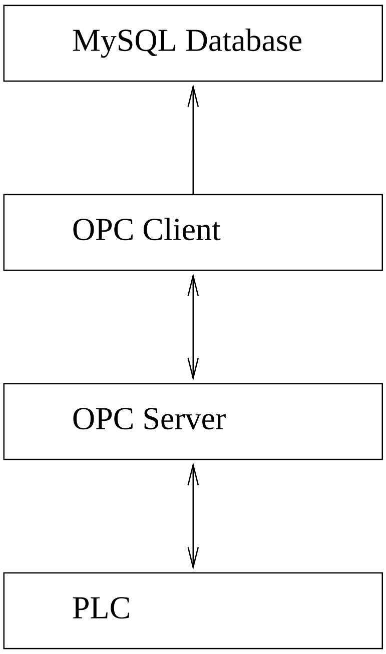


Figure 5.8: Data logging components

5.5.4 Results

Preliminary tests

To test the setup a clock pulse was monitored by the OPC server. The OPC server was set up to register each time the clock pulse changed from high to low, and low to high. The clock pulse had a frequency of 5Hz, so each cycle should last 200 milliseconds and each period should last 100 milliseconds. The result of this test is listed in table 5.5.4. The left column shows the registered period durations due to the clock pulses, and the right column is the number of times such durations were registered. As we can see there is a large variation in how long time each period is registered to be. No period is registered as to be exactly 100 milliseconds, but 97.8% lie between 93 and 110 milliseconds. The total number of periods measured are 11060, with an average duration of 99.99 milliseconds. And this is a test against a PLC which has a simple program which only stores the clock pulse in a variable.

This should give an acceptable accuracy when measuring cycle times of about 2 seconds, but if shorter times are to be measured it is important to remember this variation. It is anyway important to measure a large set of values to make sure that the average will be correct.

Period duration [ms]	Count
31.0000	15
32.0000	1
46.0000	5
47.0000	46
62.0000	45
63.0000	9
78.0000	1
93.0000	1606
94.0000	4883
109.0000	2713
110.0000	1617
140.0000	11
141.0000	53
156.0000	42
157.0000	11
172.0000	2

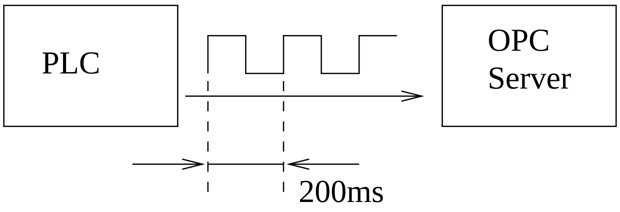


Table 5.2: PLC clock monitoring by OPC

The source of the inaccuracy is the computer. A PLC is able to set the clock with a very large accuracy, as this is a critical ability. So the computer running the OPC server, and the network must take the responsibility.

Cycle time

After testing the setup, a test run was done with the assembly cell to check the cycle time of each assembly station and subassembly. About 1500 parts were assembled in the test.

Figure 5.9 shows the variation of the cycle time in one of the assembly stations. These data are collected by the OPC server and stored by the data collection system. There is a large variance in cycle time, with two main groups at about 1.7 and 2.1 seconds. This plot shows that more work has to be done to achieve an average cycle time of 2 seconds, and to achieve a more stable cycle time. The reason why there is two peaks has to be studied more thoroughly.

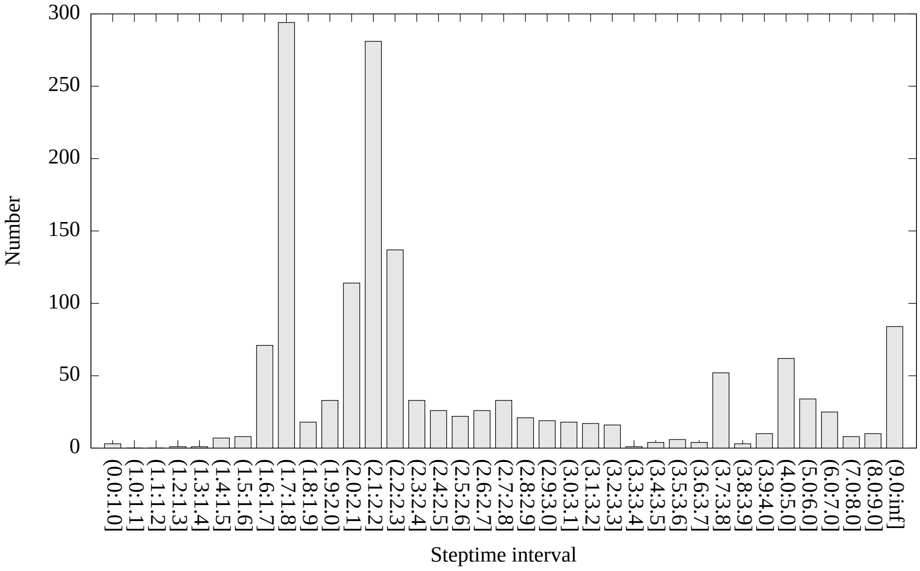


Figure 5.9: Cycle time frequency

Detailed cycle

As mentioned earlier the duration of the different states in the assembly was monitored. To get the cycle time these state times are just summarised.

Figure 5.10 shows the detailed states for some assemblies. The detailed plot is very nice for engineers who know the assembly, and it shows the PLC states very well. It is easy to spot the states which last very long, and which have a large variation.

In figure 5.11 the states are grouped into three main states. Blocking means that the pallet track is unavailable because it is indexing, Busy means that the assembly stations are doing some work on the current pallets, and Idle is when the assembly station is waiting for a new part. By grouping states like this the plot is more accessible for people without detailed knowledge, and represents well how an assembly cell works. It is important to make assembly stations non-blocking, thus making the system more robust against variations in the assembly cycles.

An interesting phenomenon is the variation of the blocking state in figure 5.11. As previously mentioned, the blocking state is the time the pallet track uses to index. The indexing is done by a screw, so the time should be constant, but it still has a variation of 0.1 second. This is the same main variation which is seen when monitoring the clock pulse of the PLC. This indicates that the best accuracy which can be achieved by OPC is 2 times the update rate.

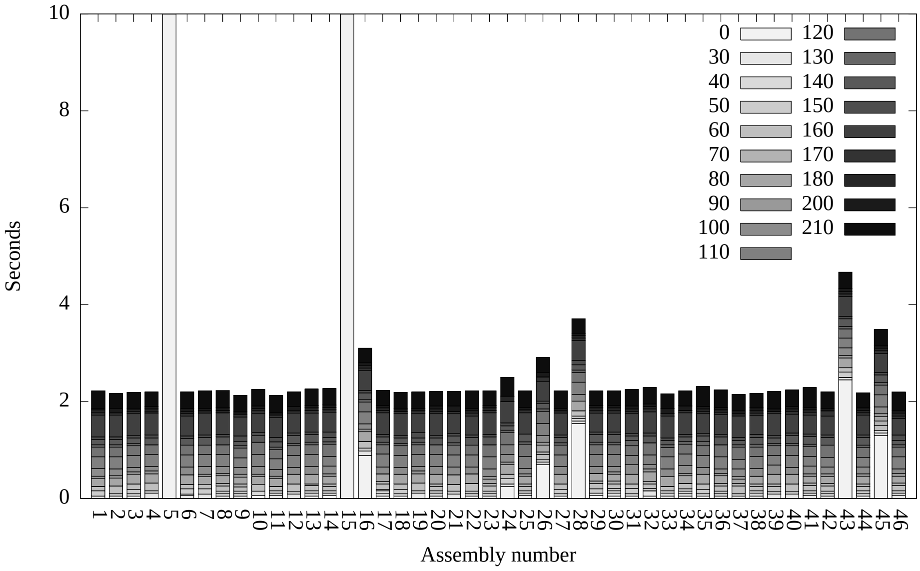


Figure 5.10: Detailed state

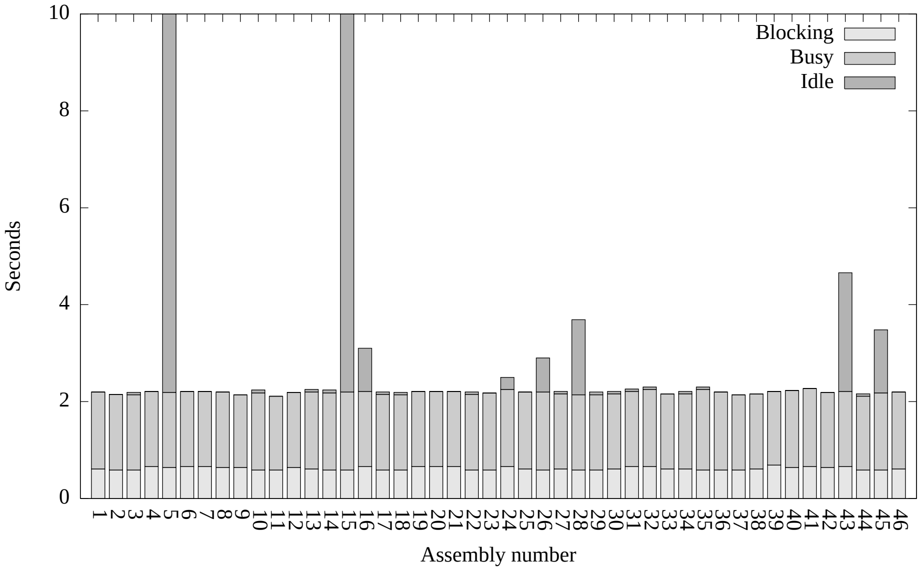


Figure 5.11: Grouped state

5.5.5 Impact on system

The state variable changes quite often, so the synchronisation between the small PLCs and the main PLC use some bandwidth on the bus. It is assumed that there is enough bandwidth at the moment, but this may change in the future. The main PLC has an add-on Ethernet module, which has its own communication processor, so the Ethernet communication should not take much resources.

In the setup the OPC server was run with an update rate of 50 milliseconds, whilst the PLC maker does not recommend a higher refresh rate than 100 milliseconds. A goal for the logging system was to have an accuracy better than 0.1 seconds, so the update rate of the OPC server had to be set higher than this. The reason for using 100 millisecond update rate is supposedly not to strain the PLC to much. The main PLC in the assembly cell does not have much work to do, as the main operational tasks were transferred to additional PLCs dedicated to different parts of the operation. So an update rate of 50 milliseconds should not be a problem.

5.5.6 Conclusions

After working with the Adept system where data collection was a flexible core functionality, using an OPC server was more complex and did not provide the same quality. The data collected did not get very accurate timestamps, and there is no guarantee that events are logged. The polling principle used by the OPC server might have been the correct principle when OPC was designed, but with the wide availability of Ethernet it is time for a standard which is based on a push principle.

In the prototype assembly cell direct monitoring of variables in the PLC program could not be done, like the status of a pallet stopper, as the variables held their value for too short a period. If a variable had a value of 0 mostly, and 1 for only a few milliseconds at a time, it would seem for the OPC server as the value always was 0. By using the state variable in the PLC program a reasonably good logging could be achieved, as the variable always has a value, and it progresses through several different values during the execution of the PLC program. Some very short states where still lost, but states which are very short are not so interesting.

The results from the logging though are very promising. Some simple results as average cycle time, and the duration of different states was quickly adopted by the engineers designing the assembly cell. With these data it was easy for them to find out where there had to be done more work to optimise the assembly cell.

Although the data is not collected with an accuracy which suits a data collection purist, the system has already been helpful and will be completed and extended as the assembly cell goes into production.

The usage of OPC made it quite simple to set up a connection to the PLC to fetch information, as it is a standard interface. But the polling principle makes it impossible to monitor I/O variables, as they often change between 0 and 1 at a high rate and there is no guarantee that they will hold their value long enough for the OPC server to notice. To make sure that interesting events are intercepted a state variable has to be introduced which keeps it value longer than the OPC polling cycle. The polling cycle is typically from 100ms and upwards. In addition the results from the OPC server will be distorted since they are not timestamped in the PLC. Network delay and the windows scheduler will both affect the accuracy of the timestamp.

Chapter 6

Summary

6.1 Introduction

In the previous chapters some thoughts and experiments regarding information collection from an assembly line have been presented. This level of information is interesting when creating an industrial production system. The information level goes down to the different elements of the process, and requires each step of the process to return a OK or FAIL result from an operation. If a more detailed level of information is necessary – to test if a single operation works correct – a different approach is needed. This will be looked into in the next part of this thesis.

With the here proposed system we get a lot of information from the production system which is of great use to engineers and operators who work with the production line on a day to day basis. But it is also possible to condense and aggregate data and make it usable by other personnel connected to the production system and in different stages of the production systems life-cycle.

6.2 Personnel

The data gathered from the production system is interesting for several different parts of the system needed for a production system to work.

- The engineer who develops the system
- The operator who maintains the system
- The planning department

- The sales department
- The purchasing department

The information needs vary a lot between the different types of personnel, but they all can get the needed information from the base data collected. It just has to be transformed into a format which is usable. The raw information is so large and complex that no one can use it directly.

As this thesis' focus is on the OEE of the production line, the work done has been focused on the needs of the engineers and operators.

6.3 Engineers needs

When designing a new production system, it is important to know the processes which are to be joined together in a line, so that the features of the processes can be used to create a stable and flexible system which in total produces with high efficiency.

An example is that a process which is very fast, but unstable, should not be joined together tightly with a slower, more stable process. The fast process should be able to build up a small buffer, which is large enough for the slower process to feed from when the fast process has to be maintained.

Realistic information about different processes can be hard to know before beginning the design, as some processes might be completely new to the designer. In addition there will always be surprises when a new process is moved from testing to production where it has to perform perfectly 24/7.

The basis of the work done in this thesis is based on the idea that a defined system can be in 4 states. It can either be processing, it can wait for parts to enter the system (it is starving), it can wait for a downstream process to accept input (it is blocked), or there can be an internal downtime. With this basis it is always possible to define a system border and to define whether a system is a bottleneck, upstream of a bottleneck, or downstream of a bottleneck.

The proposed data collection system will quickly give information about the real stability and performance for the production system, so that possible pitfalls can be noticed early in the installment phase of a new production line. Hopefully pitfalls will be identified so early that the changes required are easy to implement and do not require a major change in the system setup.

6.4 Operator needs

After the system is constructed and the engineers have done most of their work, the production system will go into a production phase.

In the production phase it should be known which critical parameters there are to watch in the future. In a complex production system there will be a lot of parameters which will vary during the lifetime of the system, and the time scope will also vary.

Some simple parameters are the stock level of components which is fed into an assembly cell. When the stock is almost depleted, it has to be refilled. Critical parameters which need instant halt and reaction is for example a crash in the system which making repairs necessary. Other interesting parameters might indicate a gradual reduction in performance. When the performance drops below a certain level, maintenance has to be done on the system.

Critical parameters, and parameters which are to be supervised by an operator are implemented in all modern production systems. Mostly there is a green/yellow/red light on a production cell which shows the current state of the system. Green light indicates that the system is working well, yellow light indicates that the system needs attention, and red indicates that a critical error has occurred. At the system there is often a computer which will show error messages, and a description of which error has to be corrected. The only problem with these systems is that they watch parameters directly in the production system without long time storage for off-line analysis.

Information which needs to be evaluated and watched over longer time periods has yet to become a standard feature in most production systems.

Part II

Force and Vibration Measurements for Error Detection

Chapter 7

Tool Condition Monitoring

Parts of this text has previously been published in the paper "Tool condition monitoring in side milling of nickel-based super alloy" by Sebastian Dransfeld, Knut Sørby and Vidar Johansen.

Monitoring tool wear and tool behaviour is important to improve the OEE of machining centres. Tool malfunction will affect all main OEE components: Availability, Performance and Quality.

Availability can be increased by preventing tool breakage, performance can be increased by finding optimal parameters which prevent excessive tool wear, and quality can be improved by changing worn tools before they affect quality.

7.1 Introduction

In machining operations of difficult-to-cut materials in the aerospace industry, the tool cost is a considerable part of the total manufacturing cost. In milling operations in nickel-based alloys like IN 100 Mod and René 77 with solid carbide end mills, the tool cost can be as high as 80% of the total cost. To keep the tool cost at an acceptable level it is necessary to use each tool as long as possible, and not change a tool that is not worn out. At the same time, a tool that is considerably worn is more likely to fail catastrophically, and tool failure is very expensive if the workpiece is damaged.

In this kind of machining operations the tool life is especially sensitive to variations in the tool geometry and other minor process variations. Therefore, the tool life cannot be predicted effectively from previous machining tests, and there is a need for process monitoring to stop the machining operation if the tool is about to fail. The milling process considered in this paper is developed as an alternative to an EDM operation. Milling is faster than EDM operations, but economical advantage can only be achieved when the milling tools are utilised efficiently, for example by use of advanced tool condition monitoring.

Common signal types used for TCM are acoustic emission ($50 - 400kHz$), vibration ($1 - 10kHz$), cutting force (static and dynamic), spindle motor torque, and spindle motor current. Acoustic emission (AE) is mainly used for detecting tool chipping or tool fracture, and for practical purposes not often used for detecting gradual tool wear [26, 31]. Some work has shown that there can be found relations between the AE signal and the tool wear [29, 61, 71], but the usefulness of AE is limited because the signal is very sensitive to noise from different sources in the machine tool system. Vibration signals are less sensitive to noise than AE, and the vibration sensors (accelerometers) and measurement system has high reliability and low cost. Similar to AE, vibration signals are used for detecting tool breakage [28], but found to be more suited than AE for monitoring tool wear [31, 63, 65]. Cutting force can be used for monitoring flank wear, either by analysis of static cutting force [48, 79], or by analysis of dynamic force [69]. In laboratory tests, measurement of cutting force is often performed by high-cost dynamometers that are very accurate, but for most practical production situations, transducers that measure the forces indirectly with limited accuracy must be used [63].

Some vendors produce monitoring systems (software and hardware) that can be integrated with the CNC system of a machine tool. These systems usually monitor cutting force or spindle torque, which are often well correlated with the degree of tool wear. Initial tests of the machining process define the levels of the acceptable force and torque. A drawback of this method is that new force and torque levels must be defined when small changes are made in the CNC program, or when tools from new suppliers are used.

The dominating wear mode for the tools considered in this work is excessive flank wear, which gives increased cutting forces and vibrations in the milling process. It is shown how vibration signal analysis in the frequency domain can be used to indicate that the flank wear criterion ($VB \approx 0.10mm$) has been reached. The result from vibration analysis is compared with results from measurements of cutting forces.

The geometric tolerance in the machining operation considered is relatively wide, and a worn tool will not spoil the geometric quality of the product. Therefore, a large tool wear can be accepted as long as the probability of catastrophic failure is sufficiently low. The usefulness of the flank wear criterion is discussed, and it is suggested that the vibration level criterion is better for predicting when the tool is about to fail.

7.2 Laboratory equipment

All machining experiments were conducted in a Bridgeport VMC 1500 vertical milling center.

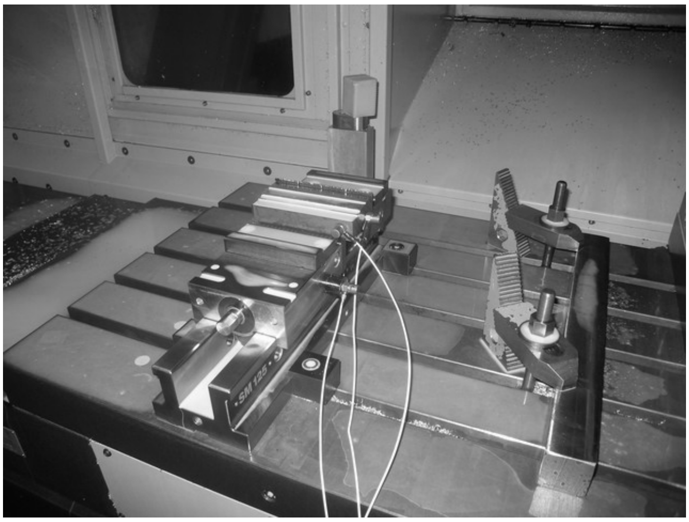


Figure 7.1: Laboratory setup

To measure forces a Kistler 9257B 3-axis dynamometer was used. The dynamometer has four sensors fitted under high preload. Each of the sensors are three-component, one component sensitive to pressure in the z direction and the other two responding to shear in the x and y directions respectively. For the forces in the x and y direction is the measure range $\pm 500N$ and in the z direction is the measurement range $\pm 1000N$. In addition the dynamometer is constructed to withstand an overload of $7.5kN$, so it is almost impossible to break it. The natural frequency of the dynamometer is $3.5kHz$, so it is important to have a lowpass filter on the charge amplifier with a lower frequency than this if dynamic forces are to be measured.

To make the signals from the dynamometer accessible a Kistler 5011 charge amplifier was used. The charge amplifier measures the charge of the dynamometer and translates it to a voltage between $0 - 10V$. The charge amplifier also has different adjustable filters.

To measure the vibrations a Kistler 8702B100 accelerometer was used. This accelerometer measures vibrations in one axis. The measurement range is $\pm 100g$, and the frequency range is $10Hz$ to $10kHz$. The resonant frequency is $54kHz$, well above the interesting measurement range. The accelerometer is connected to a Kistler 5134 coupler. The coupler supplies the accelerometer with current, it can amplify the signal from the accelerometer and it has different adjustable built-in filters. With small deflections it is important to amplify the resulting signal so that the whole voltage range is used. This improves the A/D conversion considerably.

The voltage signal from the charge amplifier and the coupler was fed into a National Instruments BNC-2110 board connected to a National Instruments DAQCard 6024E A/D converter. The DAQCard is a standard PC Card, connected to a laptop. To record and store the measured data, a LabVIEW program was written and used. The program plotted some graphs and wrote the data to disk.

During the experiments the charge amplifier and the coupler used a low-pass filter of $1kHz$, and the DAQCard sampled data at $20kHz$. The low-pass filters were well below the natural frequencies, and the sampling frequency was well above the measured frequencies.

7.3 Experiments

All experiments were conducted in co-operation with a SINTEF project which tested different end mills for Volvo Aero Norge. The purpose of the project was to measure tool wear when using different cutting parameters and different rake and clearance angles.

7.3.1 Tools

During the project 17 tools with varying geometry were tested and measured. The tests were conducted in three main series. All tools had a $6mm$ diameter, 6 teeth and a corner radius $r_\epsilon = 1.0mm$.

The first series included 3 tools, and the main focus was to check if the logging system worked and to try three different combinations of cutting speed and feed per tooth: $v_c = 40m/min, f_z = 0.1mm$, $v_c = 40m/min, f_z = 0.14mm$ and $v_c = 60m/min, f_z = 0.1mm$. After these first tests it was concluded by visual inspection of the tools that parameters $v_c = 60m/min, f_z = 0.1mm$ would lead to a rapid deterioration of the tool. The rest of the trials were conducted without using these parameters.

The next series included 9 tools. The main focus in these tests were to try different clearance and rake angles. Four different combinations were tested: $S = 8°, F = 10°$, $S = 8°, F = 14°$, $S = 12°, F = 10°$ and $S = 12°, F = 14°$. For each of these combinations the cutting parameters $v_c = 40m/min, f_z = 0.1mm$ and $v_c = 40m/min, f_z = 0.14mm$ were tried. After these tests it was concluded by visual inspection, and from the preliminary results from the force measurements, that $v_c = 40m/min, f_z = 0.1mm$ seemed to be the optimal cutting parameters.

The last series included 5 tools. These where either original Unimerco tools or tools re-grinded to have the same geometry as the original. During these tests the cutting parameters were fixed at $v_c = 40m/min, f_z = 0.1mm$, the optimum. The main object during the final series was to get more data so that it could be possible to find a relation between the measurements gathered with the optimal cutting parameters.

7.3.2 Machining

The workpiece used during all measurements was a block of IN 100 Mod, measuring $145mm \times 72mm \times 60mm$. The workpiece was mounted in a vice, which was mounted on top of the dynamometer. The dynamometer had threaded holes in it for mounting of objects. The dynamometer was mounted in a new vice which was mounted on the machine table. This setup is not optimal, and it is possible that it does not dampen vibrations as much as a specially crafted jig. The accelerometer was mounted to pick up vibrations orthogonally to the feed direction, using a specially designed magnetic holder. The tests were conducted along the $145mm$ edge.

The experiments were performed with axial depth of cut $a_p = 5.5mm$ and radial depth of cut $a_e = 0.2mm$.

The tool was taken out of the tool holder every 5 - 10 cuts for flank wear measurement in a tool makers microscope. Every cutting edge was measured, and the average flank wear was calculated. After measuring, the tool was clamped in the tool holder in the same orientation every time. All tools were used until a very large wear at the depth-of-cut line was observed, and it became clear that the tool was about to fail in a few cuts.

7.3.3 Measurements

The cuts had varying engagement time, since the cutting speed and the feed per tooth varied. It was not possible to create a connection between the milling centre and the computer, so the computer had to be operated manually. Therefor the time from the measurements began to the engagement time would vary. To remove this inconsistency and the transients that occur when the tool enters and exits the workpiece, 1 second at the beginning and at the end of the measured data was removed.

7.4 Results

7.4.1 Tool wear

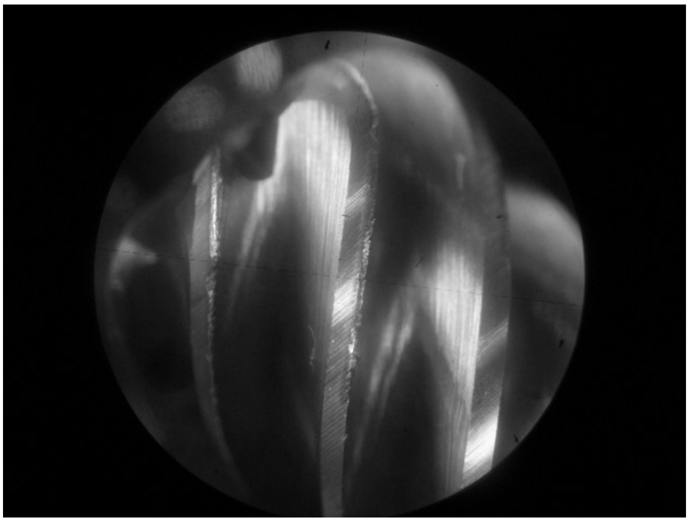


Figure 7.2: A worn tool

The tests showed that the wear on the tool edge is not uniform. As seen on the figure 7.2 the tool will have increased wear in the area which is the upper contact area with the workpiece. This increase in wear has probably two origins. The first is the increased grinding effect by the chips when they exit the tool, the other is the increased grinding effect by the build up of burr. There is also a slightly larger wear at the tool-tip, but this is expected with such a small corner radius.

On all tools that were tested during this project, the wear at the upper contact area was so large that the cutting edge in this area was completely worn out. This should increase the strain a lot at this point, and would probably be the area in which tool failure would occur.

The development in the wear on the tools cutting edges seems to happen in two distinct phases. In the first phase the cutting edges are grinded. They are reduced, but they do still look sharp. In the second phase the cutting edges break. The edges look rough and blunt, and the wear increases markable. On the tests with the highest cutting speed the phases seem to mix into each other, suggesting that it is important to not stress the tools to hard if a long life is wanted.

If we look at figure 7.3 it shows how the flank wear increases. There is first a step climb the first few cuts, this is the wear in. Then there is a stable period before the wear increases at the end. This pattern fits the standard three region tool-life curve [23, page 247]

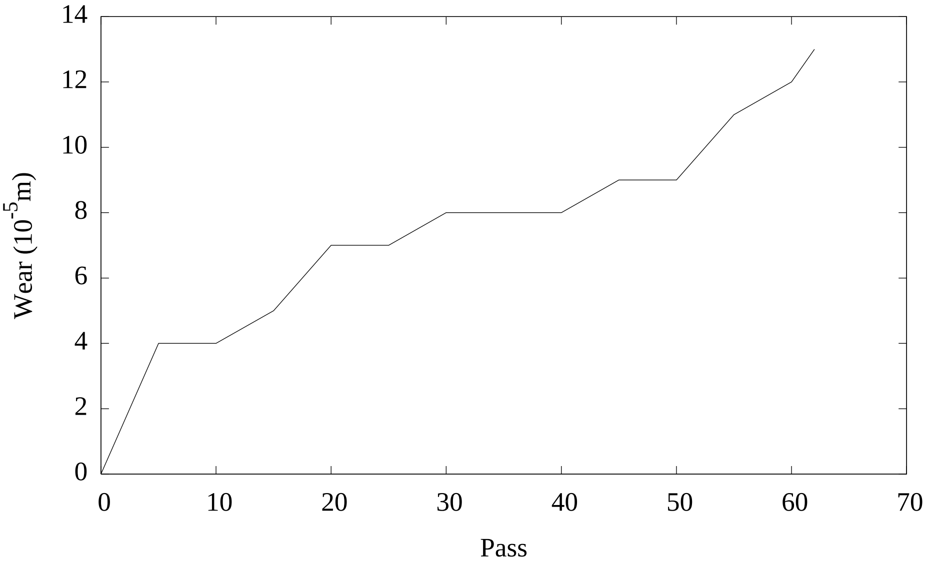


Figure 7.3: Flank wear (VB)

7.4.2 Computations

After doing the measurements the next step is to analyze the data to find trends and patterns that are equal for each tool. If there is a common factor that can be identified when a tool is worn, this can be used for tool wear monitoring.

To automate the analysis several small computer programs were made to transform the data and to create plots of the result. This made it much easier to try out a large set of different ideas.

Forces

When the tool wear increases, the resulting cutting forces will also increase. In an end milling operation with a small radial depth of cut, the force perpendicular to the feed direction will have the strongest increase. The feed force will increase to a smaller extent, and the axial force will often be approximately constant.

This is confirmed by our experiments, and the force perpendicular to the feed direction can be used as an indicator of the tool wear, see figure 7.4 for a sample. The figure shows the average of the forces measured for each cut. The usage of forces as a measurement of tool wear is a known method, and there are solution providers who use this as their method of discovery. These solution providers all have user changeable level of force

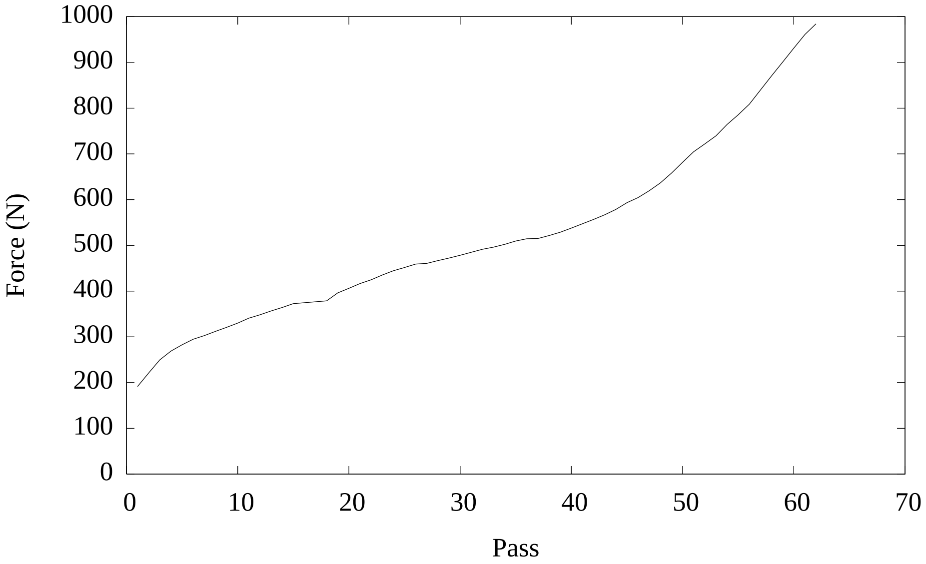


Figure 7.4: Forces perpendicular to the feed direction

increase as the point where the tool should be changed. In our experiments a level of $800N$ or an increase in the initial force by factor 4 would be adequate.

There are two interesting observations that can be made when studying the plots of force vs number of cuts. The first is that there often is high force during the first cut, before a sharp plunge. Many commercial systems use the first cut as a reference cut. Since force during the first cut (the wear in) is almost always non-consistent with the stable increase during further cuts, it might be wrong to use the first cut as a reference.

The second interesting observation is that force is non linear vs the number of cuts. Around cut 45-50 in figure 7.4 there is an increase in the slope of the curve. This increase coincides with at tool wear of about $0.10mm$. This change can be seen in all plots of force vs number of cuts. So it seems that there is a change in how the forces develop when the tool is worn out.

By doing regression analysis of measured force vs number of cuts a notable pattern can be observed when plotting the R^2 values. The first four measured values should be ignored to remove the large influence of the initial high force. There will be fluctuations in the R^2 value until the 10th cut, this is reasonable since a regression wont work well on very few values. After the 10th cut the R^2 value will be stable above 0.98, before taking a sharp plunge when the tool is worn. See figure 7.5. Some will fall to below 0.90 in a few cuts, but some only fall to 0.97, so there seems that some tools should have been tested a few more cuts. This confirms the assumption that there is a significant change in the forces

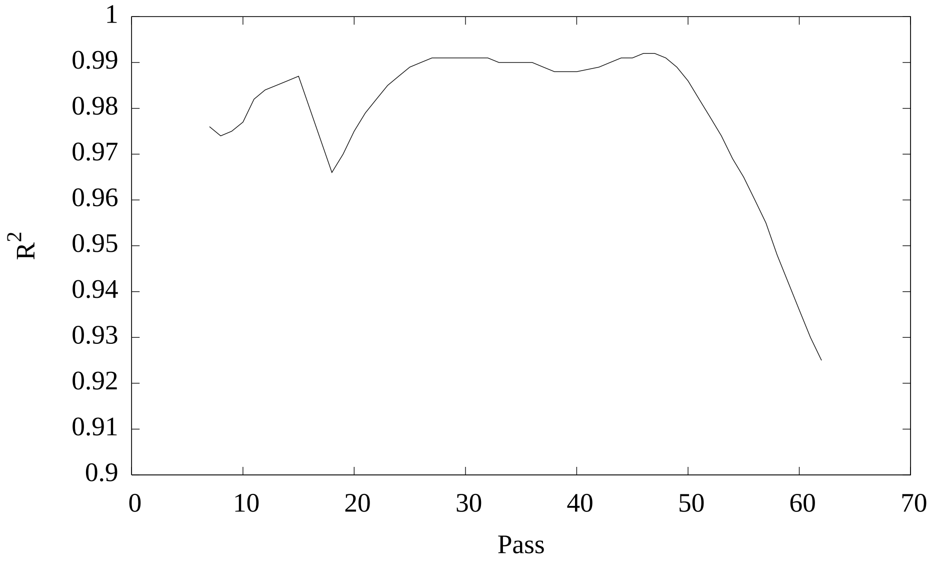


Figure 7.5: Sliding regression

when a tool is worn out.

Vibration

During milling in difficult-to-cut materials large vibrations will occur as the tool wears out. This is very easy to hear when standing next to a machine that uses a worn tool. In the tests there was found that this vibration is measurable by using an accelerometer.

The vibration data was converted into the frequency domain by computing the discrete Fourier transform (DFT) of the data set. When transformed into the frequency domain, it is possible to find out at which frequencies the main vibrations occur. To get comparable data, the results from the DFT were normalised to power spectral densities (PSD). A property of the PSD is that the square root of the area below the curve is equal to the RMS value of the signal in the time domain.

When a tool is worn and the cutting edges become increasingly blunt, the forces (and hence the forced vibrations) will increase. In the frequency domain, the changes will be noticeable in two ways:

1. The peaks at the tooth frequency and its harmonics will tend to increase
2. Small subharmonics and distortion will appear

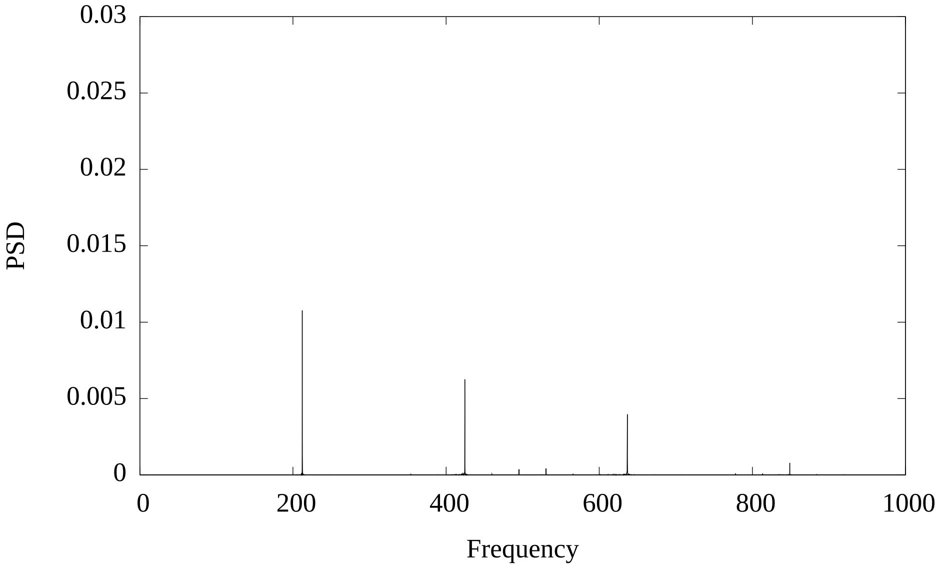


Figure 7.6: PSD for a new tool

Figure 7.6 shows the power spectrum of a new tool, and figure 7.7 shows the power spectrum of a worn tool. By comparing the spectra, a large growth in the peak at the tooth frequency and the first harmonic can be seen. In addition, a number of small peaks between the main harmonics tend to appear towards the end of the tool life. The figure for the new tool is measured during the second cut. The first cut has a completely different vibration spectrum, just as the first cut also had a much higher force. This again shows that the data measured from the first cut is not a very good reference.

The increased vibrations at the tooth frequency and its harmonics can be derived from the increased resistance from the material or the reduced cutting capacity of the tool. There will not be a recognizable pattern in the development of the peaks between the harmonics, because they represent the uneven wear of each cutting edge.

The tooth frequency in our experiments was about $212Hz$. To get the RMS value of the vibration at the tooth frequency, the area of the PSD in the range $212 \pm 15Hz$ was used. In this range the complete peak is between these bounds, but no other harmonic is included. The next harmonic is at $424Hz$, and the small peaks on both sides are at $177Hz$ and $247Hz$.

Figure 7.8 shows the development of the vibrations at the tooth frequency. All tools have a stable period before a significant increase in vibrations near the end of the tool life. The stable period might have a rising or falling trend, but the change is very small compared to the rise at the end. Because of the very sharp cutting edge for new tools, the vibrations

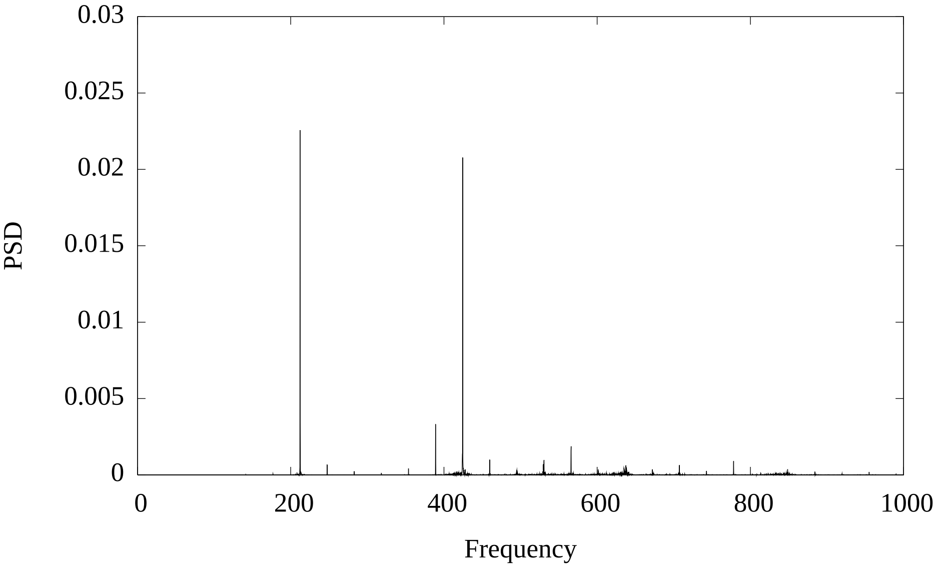


Figure 7.7: PSD for a worn tool

in the two first passes are unpredictable. Most tools will have a large value at the first pass before going into the before mentioned stable period. The flank wear is high during the first cuts, and it slows down when the stable period is reached, as shown in figure 7.3.

A problem with the growth of vibration is that there are small ups and downs all along the curve. To make a pattern that works against this a data set with the variance around a moving average of 6 samples was created. This gives another view of the vibration data gathered, and is helpful for the next section, how to monitor the tool condition.

7.5 Tool condition monitoring (TCM)

After finding properties that seem to give a good indication of tool wear, the next step is to find a method that can detect these conditions automatically. The vibrations at the tooth frequency and its multiples seems like a good candidate. Since we only have a good dataset for the tools tested with the $v_c = 40m/min, f_z = 0.1mm$ parameter set, only these are considered in this section.

The example plots shown earlier is from a tool that behaved very nice, but the data gathered from most tools varies quite a lot. So taking one tool as a reference and stating that all other tools will have the same behaviour might be a hasty conclusion. The force level is quite stable but the data gathered from the accelerometer is different for all tools. So

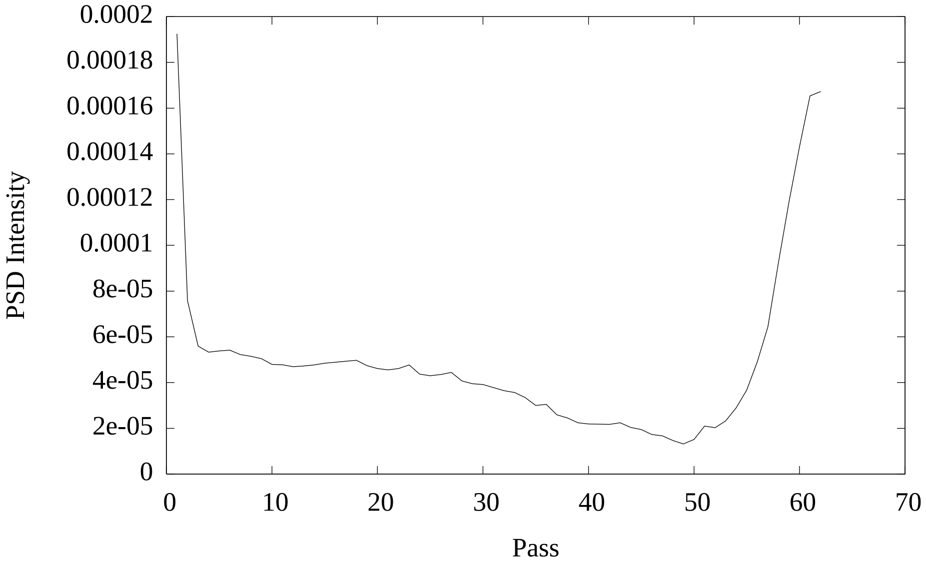


Figure 7.8: Change in the intensity of the PSD at the tooth frequency

a great supplement to the basic methods of TCM would be a method that can decide if a tool is worn just from the data gathered from previous cuts by the same tool.

Many of the features observed and commented in the previous subsection shows that there is a change in the gathered data when the tool wears out, and this change is easy to spot visually. The next step is how to detect this change automatically.

Two methods for tool wear monitoring will be proposed here. Because of the instability of the measured data it is important to have several methods to rule out false positives. The first method uses the progress of the vibrations at the tooth frequency, whilst the second method uses the progress of the variance. The objective of the methods is to be able to decide that the tool is worn by identifying the increase in signal value. By using these methods, it should not be necessary to have prior knowledge of the process. To achieve this, two preconditions are required: the data from the first pass is ignored, and the tool must survive at least 10 passes.

The first method uses the property that the slope of the curve is stable before a sudden increase. With a stable linear slope, it should be possible to use regression to predict the forthcoming values. After each pass is completed a regression is done on the measured values. The current value is checked against the upper prediction limit, and if the value is higher than this limit, an alarm is triggered. By ignoring the first measurement, which is clearly an outlier, this method gives a good result. A problem is that sometimes a small change early in the process can result in a single error.

The second method uses the property that the curve of the variance has about zero growth before a sudden increase. If the mean and the variance of these values were known, a standard control chart could be created. The sudden increase at the end would always lie above the limits that a control chart would permit. Our method uses all measured data after each pass to create a control chart, and check the current value against it. If the value lies above the upper limit, an alarm is triggered. By ignoring the first measurement, this method gives a result slightly less accurate than the previous method. The problem with this method is that when the vibrations decrease suddenly, it would result in a large variance. Though, a sudden drop in vibration is usually not a bad sign.

As mentioned earlier, each method will sometimes give false positives. Using several different methods can circumvent this. One method is to require a certain number of alarms before the process is halted, and another method is to use several systems in parallel. Using both our methods together, a good result can be achieved. Table 1 shows how many passes each tool was used before visual inspection rejected the tool, and the wear measured at the end.

Tool	Total cuts	Wear	Force		TCM	
1	62	0.13	56	90.3%	55	88.7%
2	55	0.11	50	90.9%	48	87.3%
3	50	0.13	45	90.0%	41	82.0%
4	53	0.12	47	88.7%	45	84.9%
5	68	0.12	65	95.7%	58	84.1%
6	39	0.13	33	84.6%	31	79.5%
7	40	0.23	35	87.5%	35	87.5%
8	40	0.13	31	77.5%	32	80.0%
9	35	0.14	32	91.4%	31	88.6%
10	33	0.11	26	78.8%	17	51.5%

Table 7.1: Tool life and wear

The number in the Force column shows at which pass the mean force reached $800N$, and the number in the TCM column shows at which pass both our new methods claim that the tool is worn. The percentages are based on the total amount of cuts actually achieved by each tool. The results are promising, except for tool 10, where the TCM method claims that the tool is worn too early. If tool 10 is disregarded, the statistics in Table 2 can be computed. This shows that this proposed method can compete well against a method measuring the force and comparing against a static limit.

	Force	TCM
Average	88.5%	84.7%
Maximum	95.7%	88.7%
Minimum	77.5%	79.5%
Variance	5.1%	3.6%

Table 7.2: Comparing force based halt to TCM method

7.6 Discussion

During the tests the size of the workpiece was of course reduced, but no significant shift in forces or vibrations was measured. The tests were also conducted on a block of material using a less then optimal jig. So all findings from these tests can only be taken as a hint to perform more, and preferably in real jigs used in production.

In our analysis a force criterion of $800N$ has been chosen. However, it can be discussed whether this force level is the correct one. One tool passed this level on 96% of its total number of passes, and the tool was only used for 3 more passes. In a production environment a higher safety margin would be required, so the force criterion level should probably be lowered. This could lead to rejection of several tools that are not completely worn out.

Tool number 10 was rejected too early by our proposed methods. The methods are based only on the tooth frequency, but by studying the development of the second and third harmonic of this frequency it seems that a better prediction for tool number 10 can be made.

To improve our method, it could be possible to look at the harmonics of the tooth frequency. Several of the harmonics show the same trend as the first harmonic, and in combination a better signal could be found.

After the completion of the tests described in this thesis, there has been another round of trials with the workpiece and the accelerometer mounted differently. The first results shows some deviation from the results described here, but the final result is not complete yet. This indicates that the dynamic characteristics of the mounting have effects on the results. This is as expected, and supports the idea that it is necessary with more testing, and most important, more testing with a real jig used in production.

7.7 Conclusions

Although the results from the conducted experiments are not directly usable in a production environment, they are promising. There is a shift in the data gathered from a tool when it is worn out, and it is possible to detect that. Some methods to do this have been tested, but there are still many others that could be considered.

The different analysis will probably improve as more data is gathered, and the computations are automated. One solution might be to gather different data during tests, and then display the results of different monitoring solutions to the user. Then the user could choose which solution suits best.

Even if it might be impossible to create an automated system to detect tool wear using the idea to detect the variation that will occur when a tool is worn, the analysis and methods in this chapter are still a valuable addition to previous work on tool condition monitoring. The visual representation of the data gathered will be hopefully be an interesting addition to people developing, testing, and setting up tool condition monitoring system.

As all other research done on tool wear monitoring shows, the conditions are very different between laboratory tests and a production environment. The results gathered in these experiments are promising, but several more trials have to be performed. Some trials should also be performed in a production environment to assure that the observed phenomena do not only show up in a laboratory.

Chapter 8

Force Measurement in Pick and Place Operations

8.1 Introduction

The experiments described in this chapter were conducted at NTNU in cooperation with a SINTEF project for Raufoss United, with the purpose of testing flexible assembly solutions.

Flexible assembly has been presented in section 2.2. In an OEE setting flexibility is a two-edged sword. On one side flexibility is needed to improve availability, but performance might suffer and it is easier to create good quality with special purpose systems. But with the help of advanced sensors and data analysis it is possible to create good quality, and with well designed flexible assembly cells performance will be on par with special purpose systems.

8.1.1 Background

There is a constant need for improving the efficiency and accuracy of manufacturing systems. In the automotive industry the norm for defective parts delivered to a customer is 5ppm, which is almost impossible to guarantee without very good product and process control. With a long production line there are many elements which can fail, and it is impossible to create equipment which never fails. So it is therefore necessary to test the product before it leaves the plant.

In addition to this strict quality demand, there is also a demand for higher flexibility.

There should be a constant flow of parts in a large variety, and no stock. To accomplish this the production equipment must become more flexible. Flexible machinery should be able to do similar operations, for example fitting the same part on different assemblies, or fit similar parts without shutting down for a changeover. Flexible machinery has a drawback. Production equipment created especially for one product can be custom made to do one job very well. Flexible machinery will by nature be less precise and have higher tolerances, since the machinery is designed for several different jobs. It is therefore necessary to incorporate better sensing systems to assure that an assembly has been completed correctly.

8.1.2 Testing strategies

It is impossible to produce with only 5ppm defective parts, it will be necessary to do tests in the production to assure that defective parts do not end up at the customer. There are two strategies to do this. The first strategy is to test the part after completion. The other is to test after each manufacturing operation, and assume that a part completed by a set of approved operations will comply to quality demands. Both strategies have their positive and negative sides.

To test a completed part might be difficult, since many features of the part might be hidden inside. If an error is done early in the process, several operations will afterwards be conducted on the part. These operations are all wasted since the part will be found unfit. To perform operations on a faulty part might also be dangerous to different operations along the line, since production equipment is designed to work on correct parts. There might also be a possibility to correct an error if detected early.

If each manufacturing operation is to be tested, it will require extra equipment. If testing is conducted between assembly operations it will also require both extra space and production time.

An optimal solution is to build in simple tests into the manufacturing equipment, so that errors can be detected during an operation. If something goes wrong a corrective action might be done before the part is damaged.

8.1.3 Force measurement as a way of testing

Force measurement has no tradition as a quality test in manufacturing. In the last ten years, however, force measurement has become popular in robotics, to enable robots to follow a previously unknown path and to move to an accurate position (peg-in-hole problems). If one part is rotated or positioned wrong, it is possible to correct this by force measurement. These tasks are often precision tasks, and the main focus is to do the task correctly. Often the time used is not vital.

In high-speed manufacturing, however, with steptime below 3 seconds, time is vital. In a pick and place operation it will not be enough time for a robot to move to the assembly point, do the assembly, move back for quality testing and then do corrective action. In pick and place operations with contact forces, force measurement can be an effective method for quality testing because the force can be measured during the action and corrective actions can be done instantly. Use of vision technology for this purpose is not suitable because the robot will be in the way of the camera.

Force measurement is also useful in fitting operations. If two parts are fitted together by contact forces, it might be impossible to check afterwards how the assembly operation went. The two parts might seal of against all insight, and any error will be locked up. Forces measured during the assembly might indicate if something went wrong.

Two series of experiments have been conducted to see if force measurements would be a practical tool in quality testing. In the first experiment force measurement was used to check if it is possible to determine if an assembly completed correctly, and to do corrective actions based on the information gathered. In the second experiment the focus was on detecting if an assembly and the assembly operation was completed correctly. Some tests were also done to determine the effects from different movement parameters. Correct assembly parameters for a robot are very important to find the fastest assembly speed, and to do the assembly without any excessive forces.

8.2 Equipment

The equipment used for the experiments are the same as the equipment used in chapter 5. The main properties are described in section 5.4.2. In addition a force sensor was mounted on the robot flange.

The Flexlink is not optimal for assemblies with large contact forces. The pallet is not locked well into the assembly station. On one side it is locked in its entire length, but at the other side it is only locked at the middle. This makes it possible for one corner to deflect several millimeters under pressure.

8.3 Experiments

8.3.1 First experiment

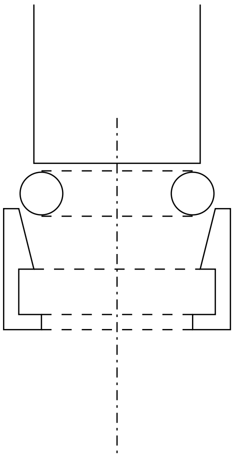


Figure 8.1: First experiment

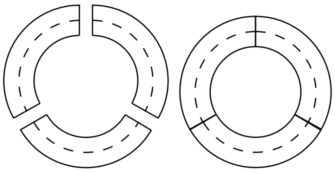


Figure 8.2: Gripper design

The first experiment was to pick up o-rings from one of the feeders and insert the ring inside a house, see figure 8.1. The o-rings were identified by vision and picked up with a specially designed tool. Then the robot moved the tool holding the o-ring to a pallet with 14 houses where the assembly was conducted. The assembly had small tolerances, so the tool had to be positioned directly above the house and centered to make the piston able to ram the o-ring into place.

The tool was an external three finger gripper which created a complete cylinder when closed, see figure 8.2. Inside this cylinder was a piston which was designed to strike the o-ring into the correct position. The size of the o-ring was bigger than the top of the house, so the tool had to compress the o-ring from $12.5mm$ to $12mm$. The piston had a diameter of $10mm$, the same diameter as the narrowest size in the house.

8.3.2 Second experiment

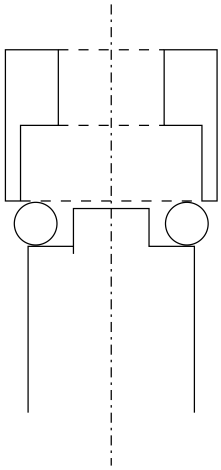


Figure 8.3: Second experiment

The second experiment was to place an o-ring inside a house. A house was picked up by a robot from one pallet, then moved to another pallet where o-rings were placed on specially crafted studs. The house was slowly pushed down onto the o-ring, then pulled back and moved back to the original pallet. The inner diameter on the stud was $0.5mm$ smaller than the inner diameter on the o-ring, and the inner diameter on the house was $0.4mm$ smaller than the outer diameter on the o-ring. The tolerances had to be so small because the stud is actually hollow because of a pipe inside the house. These small tolerances squeezed the o-ring when the house was pushed down on the stud, so it was necessary with a polished stud to make the contact forces on the house larger than on the stud.

8.4 Force measurements

On the AdeptForce controller board there is hardware which can sample the forces from the force sensor at $1kHz$. This measurement can be read from V+ by a read command, but the measured data is not available in AIM. A function was made to make the data available in AIM and then the AIM logging system was used to store the measured data, however, this was an unacceptable solution because it was too slow.

A small program was made which could run in parallel to the robot program on the controller. This program would start logging force on a signal, and print the logged data to a TCP/IP port when the logging was conducted. On the other side of the TCP/IP connection there was a computer which wrote the data to file.

The controller has one major cycle each 16ms. These 16ms are split into 16 timeslices, and the running tasks on the controller all compete for them. The standard setup allows 16 processes. A process can have priority for each of these timeslices. The trajectory calculation for the robot has the top priority on all timeslices so that the movement will work correctly. In addition the servo control loop has a special priority so that it will be run every few milliseconds.

The force logging task got a medium priority on all timeslices so that it would run quite often. Since the robot stood still, or moved one axis up or down, the force logger got a lot of CPU time. The force were read from the system, timestamped and put into a buffer. The buffer was first written out when the logging cycle had ended, thus making even more CPU time available for logging. The bulk of the readings had 0-3 millisecond between them, with 19 milliseconds in the worst recorded case. This gives a very good picture of the forces during an assembly process.

8.5 Results

8.5.1 First experiment

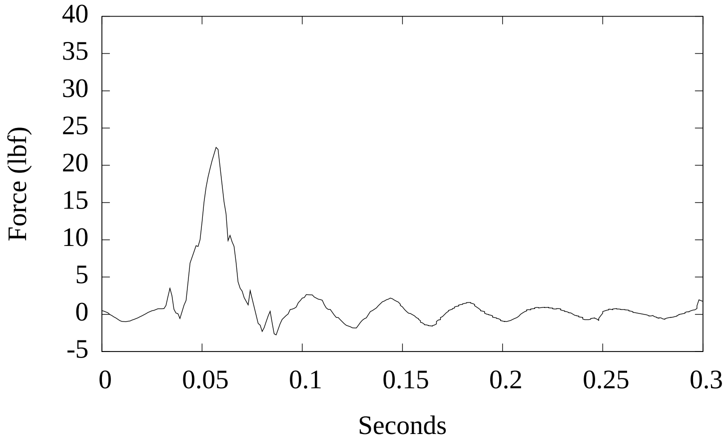


Figure 8.4: Correctly assembled o-ring

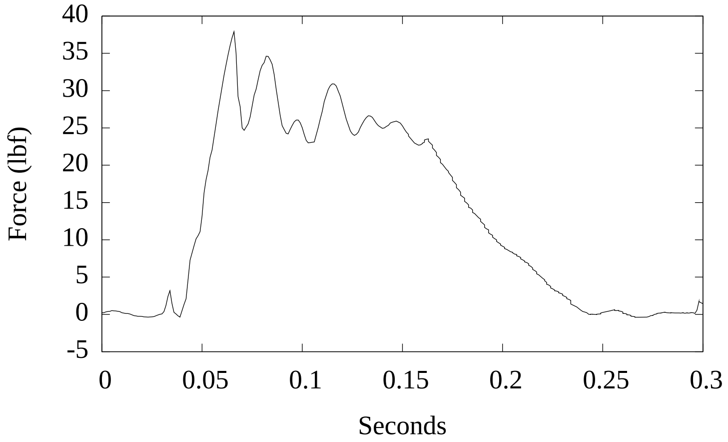


Figure 8.5: Faulty assembled o-ring

The results from the first experiment were very clear. The difference of the forces measured from a valid assembled o-ring and a faulty assembled o-ring was huge, see figures 8.4 and 8.5. The initial delay is the delay in the pneumatic system that drives the piston. The forces then go high during the downward motion. If the o-ring is assembled correctly it will place itself nicely inside the slot, and it will no longer be in contact with the piston.

If the assembly fails somehow, the o-ring will curl up and stop the piston from going to its end position.

8.5.2 Second experiment

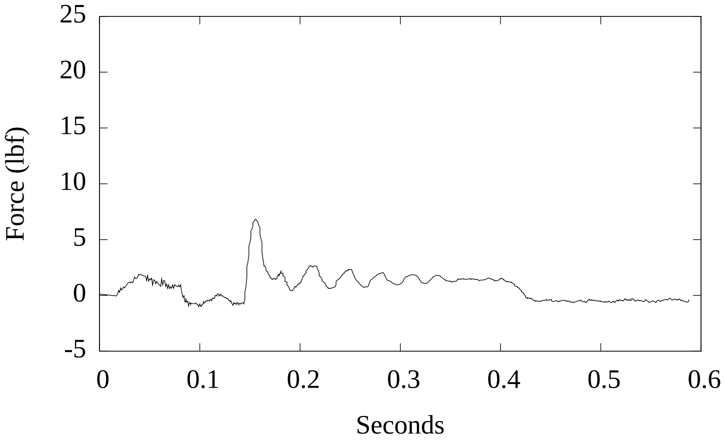


Figure 8.6: Low measured forces

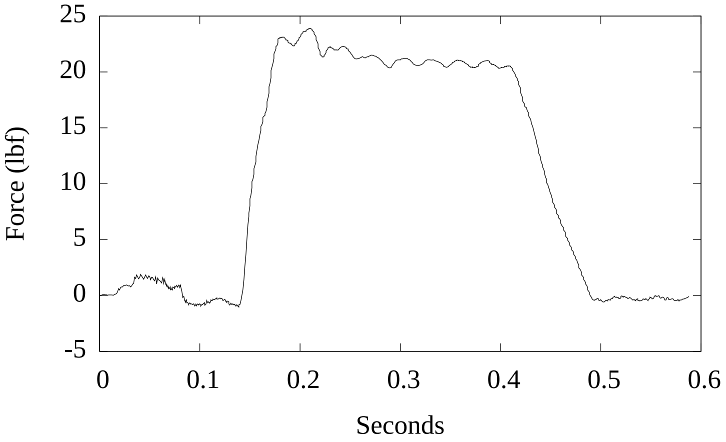


Figure 8.7: High measured forces

The results from the second experiment also looked promising, see figures 8.6 and 8.7. A problem, however, is that the force level does not tell if the o-ring is assembled correctly or not. Three states after assembly were defined: correctly assembled o-ring, o-ring is assembled but is not pushed completely in place, or the o-ring is left behind on the stud. None of these states can be deducted from the force measurements.

The cause of problem was assumed to be the friction between the stud, the o-ring, and the house. Another experiment was conducted with the o-ring sprayed with silicone. This experiment resulted in that all o-rings were assembled correctly at much lower forces.

8.6 Applicability

As seen from the previous section there is a correlation between how well a contact force assembly is conducted and the forces measured.

8.6.1 Corrective action

In the first experiment there was a significant difference between a correct and a faulty assembly. Since the o-ring still will be within the house after the piston is pulled back, it should be possible to try with several strikes to get the o-ring into position if the first strike failed.

To this purpose an algorithm was added to the force logger that would calculate if the forces collected during one stroke were too high. The algorithm had a maximum limit that the forces should not exceed during the stroke, and an average the forces should fall below after the stroke was finished. The algorithm would signal the main robot program if the assembly succeeded, and the robot program would retry up to 10 times to get the assembly to succeed.

Position	Maximum	Average	Result
1	42.17	33.36	Failed
2	39.69	32.41	Failed
3	43.49	34.80	Failed
4	36.21	24.37	Failed
5	45.34	37.95	Failed
6	12.77	0.00	Good
7	45.51	39.24	Failed
8	46.13	39.95	Failed
9	10.58	0.49	Good
10	10.56	0.51	Good
11	41.49	28.88	
12	11.12	0.34	Good
13	37.87	30.35	
14	42.14	37.25	Failed

Table 8.1: Data from one stroke

The data in table 8.1 is gathered from the first strike on one pallet. The result "Failed" is from an o-ring not correctly assembled, and the result "Good" is when the sound from the tool indicates a good assembly. In position 11 and 13 the forces are high, but the o-ring is in place. So the test is not 100% proof, but there is no harm in doing another stroke on a well assembled o-ring.

The maximum value is the maximum measured between 50 and 100 milliseconds, and the average value is the average measured between 100 and 150 milliseconds. From this, and some other measurements, a maximum limit of 20 Lb and an average limit of 1 Lb was chosen.

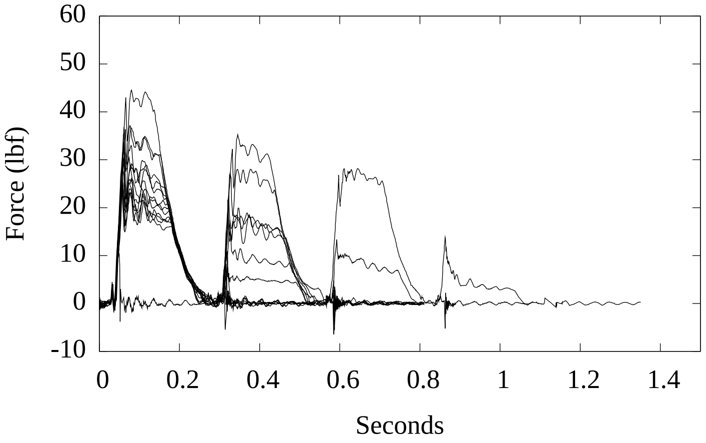


Figure 8.8: Corrective action algorithm at work

Figure 8.8 shows the gathered results from a trial run with the algorithm at work. After 5 strokes all o-rings are assembled correctly. Without the algorithm we would have conducted several unnecessary strokes which will reduce the lifetime of the tool, and increase assembly time considerably.

8.6.2 Optimization of assembly operation

In the second experiment the measured forces could not be used to tell if the assembly completed correctly or not, but it could tell if the assembly was working well. The assembly should use as low forces as possible and at the same time run at high speed.

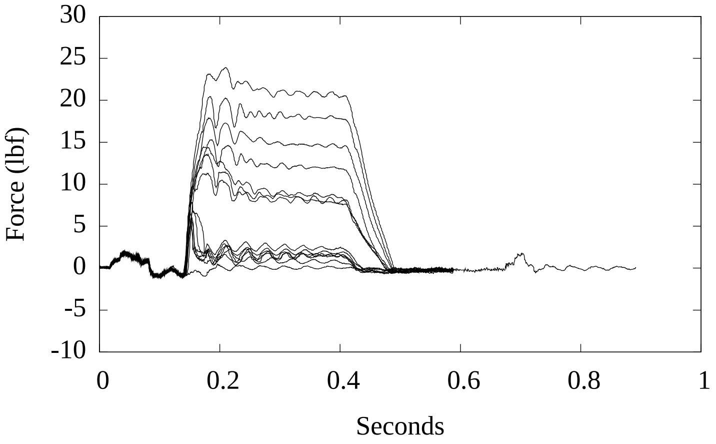


Figure 8.9: Assembly at high speed

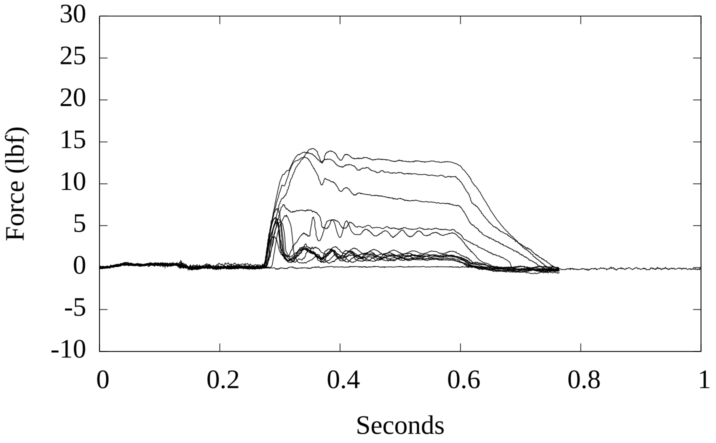


Figure 8.10: Assembly at low speed

Figures 8.9 and 8.10 show the results from doing the assembly at high speed and at low speed. Higher speed results in higher forces, and the forces vary much more. This is because of the resistance from the o-ring and the robot trying to do speed, and position control during the movement.

As mentioned earlier, there were also conducted some trials where the o-rings were sprayed with silicone. Silicone reduced the forces and increased the reliability of the assembly.

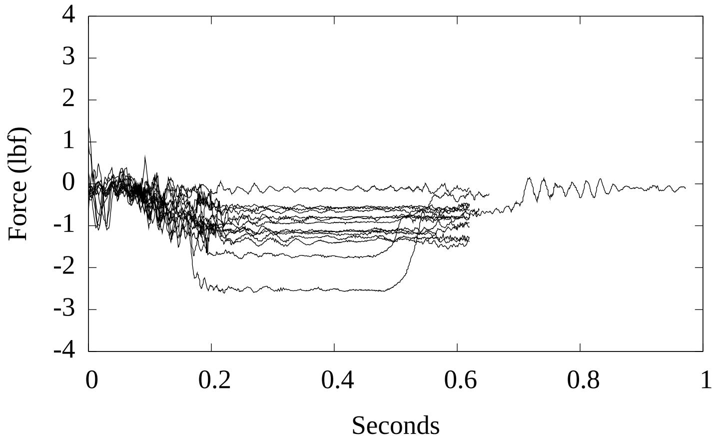


Figure 8.11: Forces sideways

The forces in figures 8.9 and 8.10 are the forces measured in the z-direction, i.e. the assembly direction. The force sensor also measured forces in the x- and y-directions. After calibrating the pallet to the robot these forces varied around 0, as seen in figure 8.11. These forces sideways are interesting in two regards. They can be used for automatically fine-tuning the position and to check if the pallet is manufactured correctly. The pallet we used had actually a flaw in one of the positions that we detected with these force measurements.

8.7 Conclusion

In an assembly where forces are expected or forces should be absent, a force sensor is a valuable tool. Force measurements will give a detailed description of what is happening during the assembly, a task that for example vision can not do. With knowledge of the forces during an assembly, we can instantly do some analysis of operations done and carry out corrective action if necessary. In high speed manufacturing this can be crucial.

Chapter 9

Summary

The usage of force and vibration data in production and assembly will give valuable information about the processes. Tool Condition Monitoring is a field which has been studied for years by many people, but more work has to done to accomplish thorough knowledge about all factors which influence toolwear.

Application of force measurements as a tool to improve assembly operations seems to work very well. Some work in this field has been done to improve precision assemblies, but force sensors are still not standard in common assembly operations.

9.1 Methodology for force measurements in assembly operations

To measure forces will contribute to an improved assembly operation, and will provide useful information for all stages in the product- and lifecycle of an assembly cell.

9.2 Initial development

During initial development force measurements will reduce time used and improve parameters used. The first stage when setting up a new assembly cell is to teach the manipulator the movement pattern it has to follow to do the assembly sequence. Since the coordinates have to be precise, the points cannot be measured and plotted into the system, the manipulator has to be moved to the different points in the assembly sequence. If the initial points are run through with a tool which has an exact fit to the assembly positions, force measurements will show which points are accurate, and which ones are not. The points can be adjusted manually or automatically until the optimal positions are found.

If pallets or other variable fixtures are used, the test tool should be used on all to check for variances between the pallets. Incorrect fixtures will lead to failures in the assembly which might be hard to track down later.

9.3 Tuning

After the complete movement path is set up, it has to be optimised. In trials speed and accelerations are increased as much as possible without jeopardising security, stability and quality. The complete assembly should also be done with predefined errors, like missing or damaged components, or misplaced components. If some scenarios indicate excessive forces, the assembly has to be monitored to avoid such situations.

9.4 Quality control

After the development phase force measurements can be used during assembly operation for quality control. If force measurements have been used already during development, the force picture for different scenarios should be known. Any forces deviating to much from expected will indicate that there has been a problem during assembly. Studying such deviations should enable to create a knowledge database that can be used to understand registered force patterns and their meaning.

9.5 Adaptive assembly

A good correlation between a deviating force pattern and an error during assembly might be used to create an adaptive assembly. The main requirement to do this is that the operation concerned can be redone. Most important is to avoid damage on the main object which the assembly operation is conducted on. If some assembly part is damaged, it will be possible to remove this, and try with a new one.

Two methods of creating an adaptive assembly are discussed; either go through with the complete assembly operation, check the results, and then try again if it fails; or be careful during the complete assembly operation and check at each step if everything is OK, correcting any deviation detected.

Complexity, available time, and product cost will decide if adaptive assembly is feasible. A complex operation, which might require a tool change, on a cheap part would not be cost effective. A simple operation, however, fixing an inept operation threatening an expensive part should be welcome.

Part III

User Interface

Chapter 10

User Interface

10.1 Introduction

In chapter 6 the data needs from a production system was discussed. In this chapter the details of how the data presentation can be implemented will be discussed.

When a new production line is engineered it is important for the engineer to get quick access to the data, with a user interface that gives the user possibilities.

When the production line is completed and set into production a simpler user interface needs to be created, which can show a set of predefined views of the data. This interface should have a general front end which displays the data needed for day to day surveillance, but it should also be possible to delve deeper into the system.

In this chapter two systems for creating user interfaces will be presented.

The first is the spreadsheet. Spreadsheets are well designed for manipulating data, but it is difficult to create a good user interface which is fool proof. This makes it a good tool for an engineer which works with off-line data. The possibilities are many, but it requires a lot of training.

The second is the web interface. To create a web interface can be simple with modern tools, but it still requires some knowledge to set up. Quite complex user interfaces can be created, but the web interface should only be preferred when creating simple interfaces which are mostly read only.

10.2 Spreadsheets

Graphs and tables are the speciality of modern spreadsheets. This makes them a good Rapid Application Development (RAD) tool for data analysis work. It is easy to export data from the SQL (Structured Query Language) database to a spreadsheet, either by dumping the database or using a SQL-interface from the spreadsheet. Since the data has to be exported from the database, the spreadsheet is not a very good tool for online data visualisation.

After the data is imported into the spreadsheet, the numerical functions make it easy to analyse the results and to make a good presentation. One main limitation of a spreadsheet is the low number of possible rows. A dataset from a data warehouse might consist of millions of rows, but a spreadsheet can usually only handle some thousand rows.

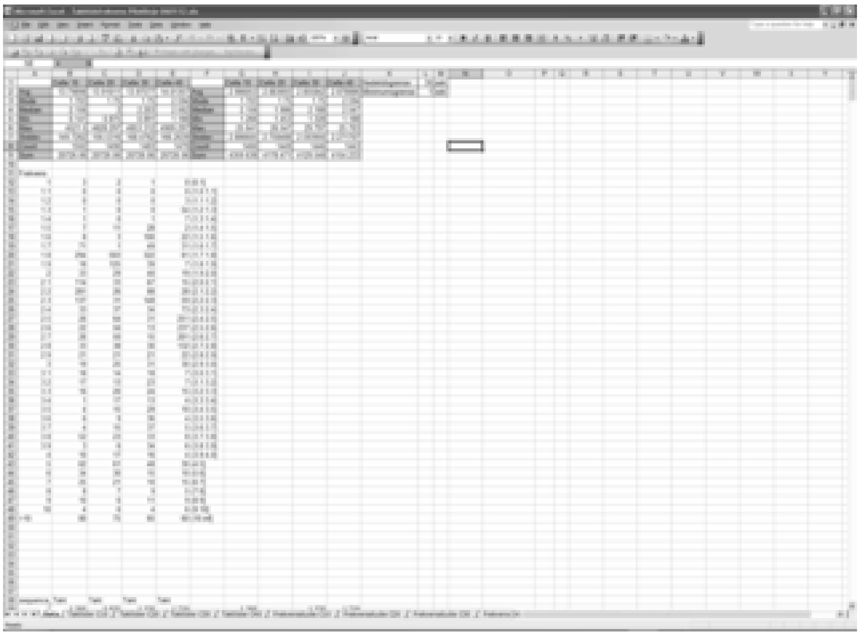


Figure 10.1: Spreadsheet data

As can be seen in the figures 10.1 and 10.2 the visual representation of the data in a spreadsheet is good, and it is not very difficult to make it work.

The problems with the spreadsheet approach is that the spreadsheet cannot handle very large amounts of data and there is limited support against online data warehouse.

For a system developer the spreadsheet approach will work quite well as it is flexible, and changes can be done without too much effort.

Figure 10.2: Spreadsheet chart

A spreadsheet will therefor be valuable during the first period of development of a new production line.

10.3 Web interface

Parallel with the completion of a production line, the data needed to monitor the data to data operation of a production line should be defined. The data needed should be based on the ideas and principles of OEE.

Since there should be no need to edit or change the data after the production line has been completed, a web interface can be used to create the interface. Web interfaces are very popular today, as most modern computers are equipped with a web browser. There is no need to install anything, and the data can be accessed from anywhere in the world.

Many of the elements from the spreadsheets can be imported into the web interface, so the ideas and the experience from the spreadsheet can be used when designing the web interface.

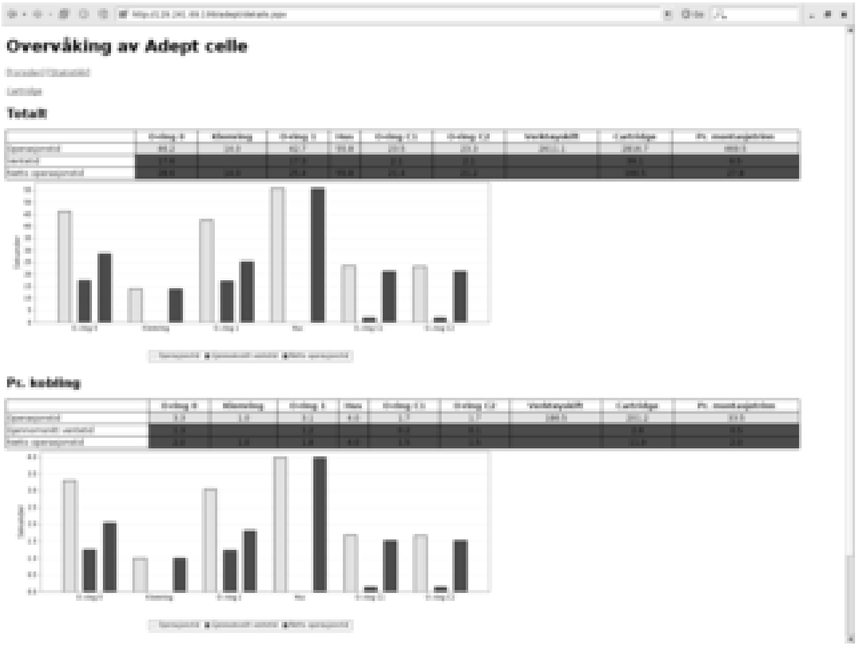


Figure 10.3: Screenshot from the web presentation

Figure 10.3 is a screenshot from a simple web page made for this project. A web server with necessary programs and libraries were installed on the logging computer, and some basic statistics for the first experiment was presented. The website was run by Tomcat ([72]), and the graphics was created by using JFreeChart ([41]). It was also possible to get a dump of the data, that could be imported into a spreadsheet.

10.4 Conclusions

It is important to fetch data from a production line early during development, but which data to fetch and how to present them might not be clearly defined. A RAD tool should be used to both enable the engineer to view data and to gather experience for the final user interface.

The final user interface should be easily accessible, preferably by remote computers, and give a good overview of day-to-day key performance indicators.

The input data to a user interface might vary a bit from system to system, but the main views will be similar. So by creating data accessors which fetches data from the dataset, and standard view components that interacts with the data accessors through a standard interface, it will be possible to reuse many components from one system to another.

During the project user interfaces was created for all production setups. For the prototype line at Raufoss United the database and user interface gave a good insight in how the production line functioned, and the company who engineered the line used it to improve it. The system has been continued and is now connected to the resulting production line.

Bibliography

[1] I. Abu-Mahfouz. Drilling wear detection classification using vibration signals and artificial neural network. *International Journal of Machine Tools & Manufacture*, 43:707–720, 2003.

[2] Adept Technology, Inc. AdeptForce VME User's Guide, 1995.

[3] Adept Technology, Inc. AdeptNet User's Guide, 1996.

[4] Adept Technology, Inc. Adept MV Controller User's Guide, 1999.

[5] Adept Technology, Inc. Instruction Handbook: Adept Cobra 600 and Adept Cobra 800 Robot, 1999.

[6] Adept Technology, Inc. Adept FlexFeeder 250 User's Guide, 2000.

[7] Adept Technology, Inc. AIM Baseline User's Guide, 2000.

[8] Adept Technology, Inc. AIM Customizer's Reference Guide, 2000.

[9] Adept Technology, Inc. MotionWare Reference Guide, 2000.

[10] Adept Technology, Inc. MotionWare User's Guide, 2000.

[11] Adept Technology, Inc. V+ Language Reference Guide, 2000.

[12] Adept Technology, Inc. V+ Language User's Guide, 2000.

[13] Adept Technology, Inc. V+ Operating System Reference Guide, 2000.

[14] Adept Technology, Inc. V+ Operating System User's Guide, 2000.

[15] Adept Technology, Inc. VisionWare Reference Guide, 2000.

[16] Adept Technology, Inc. VisionWare User's Guide, 2000.

[17] Adept Technology, Inc. AdeptVision Reference Guide, 2001.

[18] Adept Technology, Inc. AdeptVision User's Guide, 2001.

[19] W. Amer, R. I. Grosvenor, and P. W. Prickett. Sweeping filters tooth rotation energy estimation (tree) techniques for machine tool condition monitoring. *International Journal of Machine Tools & Manufacture*, 46:1045–1052, 2006.

[20] T. Arai, N. Yamanobe, Y. Maeda, H. Fujii, T. Kato, and T. Sato. Increasing efficiency of force-controlled robotic assembly – design of damping control parameters considering cycle time –. In *Annals of the CIRP*, volume 55/1, pages 7–10, 2006.

[21] Artis. `http://www.artis.de/`.

[22] M. Balazinski, E. Czogalab, K. Jemielniakc, and J. Leski. Tool condition monitoring using artificial intelligence methods. *Engineering Applications of Artificial Intelligence*, 15:73–80, 2002.

[23] Beno Benhabib. *Manufacturing: Design, Production, Automation, and Integration*. Marcel Dekker, Inc., 2003.

[24] Blum-novotest. `http://www.blum-novotest.com/`.

[25] Brankamp. `http://www.brankamp.com/`.

[26] G. Byrne, D. Dornfeld, I. Inasaki, G. Ketteler, W. König, and R. Teti. Tool condition monitoring (TCM) – the status of research and industrial application. In *Annals of the CIRP*, volume 44/2, pages 541–567, 1995.

[27] G. Chand and B. Shirvani. Implementation of TPM in cellular manufacture. *Journal of Materials Processing Technology*, 103:149–154, 2000.

[28] J. C. Chen and W. Chen. A tool breakage detection system using an accelerometer sensor. *Journal of Intelligent Manufacturing*, 10:187–197, 1999.

[29] K. T. Chung and A. Geddam. A multi-sensor approach to the monitoring of end milling operations. *Journal of Materials Processing Technology*, 139:15–20, 2003.

[30] George Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed Systems: Concepts and Design*. Pearson Education, third edition, 2001.

[31] D. E. Dimla. Sensor signals for tool-wear monitoring in metal cutting operations – a review of methods. *International Journal of Machine Tools & Manufacture*, 40:1073–1098, 1999.

[32] R. Du and K. Yeung. Fuzzy transition probability: a new method for monitoring progressive faults. part 1: the theory. *Engineering Applications of Artificial Intelligence*, 17:457–467, 2004.

[33] R. Du and K. Yeung. Fuzzy transition probability: a new method for monitoring progressive faults. part 2: Application examples. *Engineering Applications of Artificial Intelligence*, 19:145–155, 2006.

[34] K. Feldmann and S. Slama. Highly flexible assembly – scope and justification. In *Annals of the CIRP*, volume 50/2, pages 489–498, 2001.

[35] N. Ghosh, Y. B. Ravi, A. Patra, S. Mukhopadhyay, S. Paul, A. R. Mohanty, and A. B. Chattopadhyay. Estimation of tool wear during cnc milling using neural network-based sensor fusion. In press, 2005.

[36] M. Hägele, W. Schaaf, and E. Helms. Robot assistants at manual workplaces: Effective co-operation and safety aspects. In *Proceedings of the 33rd ISR*, 2002.

[37] B. Huanga, H. Goua, W. Liua, Y. Lia, and M. Xie. A framework for virtual enterprise control with the holonic manufacturing paradigm. *Computers in Industry*, 49:299–310, 2002.

[38] S. Jack Hu and J. Camelio. Modeling and control of compliant assembly systems. In *Annals of the CIRP*, volume 55/1, pages 19–22, 2006.

[39] J. Jarvisa, R. Rönnquista, D. McFarlaneb, and L. Jain. A team-based holonic approach to robotic assembly cell control. *Journal of Network and Computer Applications*, 29:160–176, 2006.

[40] Java. `http://java.sun.com/`.

[41] JFreeChart. `http://www.jfree.org/jfreechart/`.

[42] S. M. Ji, L. B. Zhang, J. L. Yuan, Y. H. Wan, X. Zhang, L. Zhang, and G. J. Bao. Method of monitoring wearing breakage states of cutting tools based on mahalanobis distance features. *Journal of Materials Processing Technology*, 129:114–117, 2002.

[43] F. Jovane, Y. Koren, and C. R. Boër. Present and future of flexible automation: Towards new paradigms. In *Annals of the CIRP*, volume 52/2, pages 543–560, 2003.

[44] J. Krüger, R. Bernhardt, and D. Surdilovic. Intelligent assist systems for flexible assembly. In *Annals of the CIRP*, volume 55/1, pages 29–32, 2006.

[45] J. L. M. Lastraa and A. W. Colombo. Engineering framework for agent-based manufacturing control. *Engineering Applications of Artificial Intelligence*, 19:625–640, 2006.

[46] Lean manufacturing. `http://en.wikipedia.org/wiki/Lean_Manufacturing`.

[47] P. Leitão and F. Restivo. Adacor: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, 57:121–130, 2006.

[48] S. C. Lin and R. J. Yand. Force-based model for tool wear monitoring in face milling. *International Journal of Machine Tools & Manufacture*, 35/9:1201–1211, 1995.

[49] Linux. `http://www.kernel.org/`.

[50] N. P. Mahalik, editor. *Fieldbus Technology, Industrial Network Standards for Real-Time Distributed Control*. Springer, 2003.

[51] H. Makino and T. Arai. New developments in assembly systems. In *Annals of the CIRP*, volume 43/2, 1994.

[52] K. Martinsen and G. Knutstad. Socio-technical aspects of machine tool process monitoring. In *36th CIRP International Seminar on Manufacturing Systems Proceedings*, pages 571–578, 2003.

[53] M. Meng, C. Chen, P. X. Liu, and M. Rao. E-service robot in home healthcare. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 832–837, 2000.

[54] G. F. Micheletti, W. König, and Victor H. R. In process tool wear sensors for cutting operations. In *Annals of the CIRP*, volume 25/2, pages 483–496, 1975.

[55] Sape Mullender, editor. *Distributed Systems*. Pearson Education, second edition, 1993.

[56] MySQL. `http://www.mysql.com/`.

[57] Issa A. D. Nesnas. Computer vision strategies for flexible part feeding. Robotics Online `http://www.roboticsonline.com/`, 1997.

[58] D. Noe. Sensing and acting in intelligent assembly systems. *Annual Reviews in Control*, 23:171–176, 1999.

[59] Ntp. `http://www.ntp.org/`.

[60] Richard Oechsner, Markus Pfeffer, Lothar Pfitzner, Harald Binder, Eckhard Müller, and Thomas Vonderstrass. From overall equipment efficiency (OEE) to overall fab effectiveness (OFE). *Materials Science in Semiconductor Processing*, 5:333–339, 2003.

[61] R. H. Osuri, S. Chetterjee, and S. Chandrashekhar. On-line condition monitoring of tool wear in end milling using acoustic emission. *International Journal of Production Research*, 29/7:1339–1353, 1991.

[62] Overall equipment effectiveness. `http://en.wikipedia.org/wiki/OEE`.

[63] P. W. Prickett and C. Johns. An overview of approaches to end milling tool monitoring. *International Journal of Machine Tools & Manufacture*, 39:105–122, 1999.

[64] U. Rembold, B. O. Nnaji, and A. Storr. *Computer Integrated Manufacturing and Engineering*. Addison-Wesley Publishing Company, 1993.

[65] J. Rotberg, E. Lenz, and S. Braun. Mechanical signature analysis in interrupted cutting. In *Annals of the CIRP*, volume 36/1, pages 249–252, 1987.

[66] O. Ryabov, K. Mori, and N. Kasashima. An in-process direct monitoring method for milling tool failures using a laser sensor. In *Annals of the CIRP*, volume 45/1, pages 97–100, 1996.

[67] D. R. Salgado and F. J. Alonso. Tool wear detection in turning operations using singular spectrum analysis. *Journal of Materials Processing Technology*, 171:451–458, 2006.

[68] M. Santochi and G. Dini. Sensor technology in assembly systems. In *Annals of the CIRP*, volume 47/2, pages 503–524, 1998.

[69] A. Sarhan, R. Sayed, A. A. Nassr, and R. M. El-Zahry. Interrelationships between cutting force variation and tool wear in end-milling. *Journal of Materials Processing Technology*, 109:229–235, 2001.

[70] M. Sortino. Application of statistical filtering for optical detection of tool wear. *International Journal of Machine Tools & Manufacture*, 43:493–497, 2003.

[71] P. Srinivasa Pai, T. N. Nagabhushana, and P. K. Ramakrishna Rao. Tool wear estimation using resource allocation network. *International Journal of Machine Tools & Manufacture*, 41:673–685, 2001.

[72] Tomcat. `http://tomcat.apache.org/`.

[73] H. K. Tönshoff, J.P. Wulfsberg, H. J. J. Kals, W. König, and C. A. van Luttervelt. Development trends in monitoring and control of machining processes. In *Annals of the CIRP*, volume 37/2, pages 611–622, 1988.

[74] E.H. Van Leeuwen and D.H. Norrie. Intelligent manufacturing: holons and holarchies. *Manufacturing Engineer*, 76(2):86–88, 1997.

[75] A. Velayudham, R. Krishnamurthy, and T. Soundarapandian. Acoustic emission based drill condition monitoring during drilling of glass/phenolic polymeric composite using wavelet packet transform. *Materials Science Engineering A*, 412:141–145, 2005.

[76] W. Wang, Y. S. Wong, and G. S. Hong. Flank wear measurement by successive image analysis. *Computers in Industry*, 56:816–830, 2005.

[77] H.J. Warnecke. Assembly systems now and in the future research and development. In *Annals of the CIRP*, volume 27/2, pages 597–605, 1978.

[78] A. Watanabe, S. Sakakibara, K. Ban, M. Yamada, and G. Shen. A kinematic calibration method for industrial robots using autonomous visual measurement. In *Annals of the CIRP*, volume 55/1, pages 1–6, 2006.

[79] W. Yan, Y. S. Wong, K. S. Lee, and T. Ning. An investigation of indices based on milling force for tool wear in milling. *Journal of Materials Processing Technology*, 89-90:245–253, 1999.