MASTER

Development of a binary process quality control methodology

Peters, H.P.A.

Award date:
2006

Development of a binary
Process Quality Control methodology

Drs. H.P.A. Peters

Master's thesis

Supervisor: Prof. dr. ir. J.E. Rooda
Coaches: Dr. A.Y. Pogromsky
Ir. J.G. van Reede (Philips Lighting, Roosendaal)

EINDHOVEN UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING
SYSTEMS ENGINEERING GROUP

Amsterdam, February 2006

Preface

In the beginning of 2005 Jan van Reede of the Global Services division of Philips Lighting approached me with an interesting opportunity. He was investigating the monitoring of the process of incandescent lamps, a product of which Philips sells over 2 billion pieces every year.

Only recently the sensor data has been extracted from the control system and therefore has become available for analysis. The initial question was very broad: What can we do with this data? It can be presented in several ways, but how should it be analyzed? What underlying information can be surfaced? Is it possible to predict which problems are present in the line? Maybe even which root-causes have led to these problems?

Answering these questions has proved to be an extremely exciting exercise. Obviously, the training and testing of the developed tool in the beautiful country of Indonesia have contributed in this respect. A very interesting experience there has been the implementation of a system, developed by Jan van Reede which presents the reject levels of all sensors to the operators on a real-time basis.

I would like to thank Sasha Pogromsky for his insightful comments during my final project and my fellow student Luuk van Laake for reviewing several parts. Finally, many thanks to Jan van Reede for our brainstorm sessions, and his pleasant company on different small islands around Indonesia.

Harm Peters

Amsterdam, February 2006

Summary

An interesting topic in the field of improvement of manufacturing processes is formed by process quality control. It aims to improve quality and productivity by identifying root-causes of problems in the production line. Only for the most advanced industries such as the wafer and automobile industry effective PQC methodologies have been developed [Tob97, Apl01]. For many other industries, that use binary sensors to monitor the production process, these methodologies can not be applied however.

The aim of this research project has been to fill this void by developing a binary methodology. Hereto, the applicability of continuous techniques has been thoroughly examined. The inherently lower level of information in binary sensor data appears to exclude the possibility of modeling the relationship between root-causes and sensor data directly. Therefore, indirect classification techniques that are based on training need to be used.

First, an analysis tool has been developed that is able to characterize each production period as normal or abnormal, given its binary sensor data. Hereto, a calculation process termed *entitlement calculation* has been introduced. The abnormalities are then included as binary features in the description of the system state. Several other features, such as correlations relevant from an engineering point of view and events as change overs can be added.

Next, several classification algorithms were developed and tested on the practical case of producing incandescent lamps at Philips Lighting. These algorithms are able to compare new cases to diagnosed training cases, and as such assess which situation in the past is most similar to the current system state. In order to diagnose the training cases a camera tool has been developed that is able to take pictures of rejects at 100 positions before and after each operation. It is assumed that the root-causes present in the current case correspond to the causes in the most similar training case.

The methodology has been elaborately tested for Philips and yielded very good classification results, varying between 80% and 100%. As current waste levels per production line account for 90k euro of loss per year [Ree05], and over 200 production lines exist around the world, millions of euros of rejects are thrown away. If informing the operators, technicians and engineers on the present root-causes of problems leads to only percentages of efficiency improvement, savings could be substantial. The recommendation is, however, to first extend the scope and functioning of the tool.

Samenvatting (in Dutch)

Een interessant aandachtsveld op het gebied van verbetering van productiesystemen is *process quality control.* PQC methodes hebben als doel het verbeteren van kwaliteit en productiviteit middels het vaststellen van de hoofd-oorzaken van problemen die in het proces aanwezig zijn. Enkel voor de meest geavanceerde productie processen, zoals de wafer en automobiel industrie, zijn dergelijke methodes in uitgebreide vorm ontwikkelt [Tob97, Apl01]. Het blijkt echter niet mogelijk deze methoden direct te gebruiken voor andere processen, waar vaak gebruikt wordt gemaakt van binaire sensoren.

Het doel van dit onderzoeksproject is het ontwikkelen van een PQC methodologie voor de binaire industrie. De toepasbaarheid van de continue methodes is allereerst grondig onderzocht. Aangezien binaire sensor data minder informatief is dan continue data, blijkt het niet mogelijk de relatie tussen de hoofd-oorzaken van problemen en de sensor data op een directe manier te modelleren. Om die reden dient uitgeweken te worden naar classificatie technieken welke gebaseerd zijn op training.

Als eerste stap is een analyse tool ontwikkelt welke, gegeven de binaire sensor data, elke productie periode als normaal of abnormaal kan bestempelen. Het concept *entitlement calculation* is hiertoe geintroduceerd. De abnormale datapunten worden daarna meegenomen als binaire eigenschappen van de status van het systeem. Andere systeem eigenschappen, zoals het optreden van specifieke correlaties die vanuit de kennis van het productie proces relevant worden geacht, of het optreden van gebeurtenissen als omstellingen kunnen worden toegevoegd.

Vervolgens zijn verschillende classificatie algoritmes ontwikkeld en getest voor de productie van gloeilampen bij Philips Lighting. Deze algoritmes vergelijken nieuwe cases met *training cases*, en schatten op grond daarvan in op welke situatie uit het verleden de huidige case het meest lijkt. Een camera applicatie is ontwikkeld om de training cases te kunnen diagnosticeren op aanwezige hoofd-oorzaken van problemen. De applicatie kan in een range van 100 posities voor en na elke bewerking foto's van de rejects maken. Hierbij wordt aangenomen dat de hoofd-oorzaken aanwezig voor het huidige geval gelijk zijn aan die in de meest overeenkomende training case.

De methodologie is uitgebreid getest voor Philips met zeer goed resultaat. Het percentage correcte classificaties varieert tussen de 80% en 100%. Aangezien de huidige afval niveau's circa 90k euro per lijn per jaar bedragen [Ree05], en er meer dan 200 productie lijnen over heel de wereld zijn, wordt jaarlijks voor miljoenen euro's aan rejects weggegooid. Als het informeren van operators, technici en engineers omtrent de voorspelling van aanwezige hoofd-oorzaken van problemen ook maar tot enkele procenten verbetering kan leiden, dan kunnen de besparing substantieel zijn. De aanbeveling is echter om de methodologie eerst nog verder uit te breiden.

Contents

Preface i

Summary iii

Samenvatting (in Dutch) v

List of Symbols xi

1 Introduction 1

1.1 Objective 2

1.2 Approach 2

2 Process Quality Control 5

2.1 Process Quality Control loop 5

2.2 Categorization by production process 7

2.3 Summary 9

3 Review of continuous methodologies 11

3.1 Monitoring techniques 11

3.1.1 Effectiveness of monitoring 13

3.2 Analysis techniques 13

3.2.1 State Space Modeling 13

3.2.2 Spatial Signature Analysis 15

3.2.3 Diagnostic Feature Extraction with the use of Principal Component Analysis 17

3.2.4 Statistical Process Control 20
3.3 Classification techniques 22
3.3.1 Direct techniques 23
3.3.2 Semi-direct techniques 23
3.3.3 Indirect techniques 23
3.4 Summary 26

4 Development of binary methodology **27**
4.1 Monitoring with binary sensors 28
4.2 Applicability of continuous techniques 29
4.2.1 Analysis techniques 29
4.2.2 Classification techniques 30
4.3 Analysis technique development 31
4.3.1 Entitlement calculation 31
4.3.2 Fitting a distribution 39
4.3.3 Binomial and Poisson distribution 41
4.3.4 Entitlement check 45
4.4 Classification technique development 47
4.4.1 Binary feature selection 47
4.4.2 Learning algorithms 49
4.4.3 Training the tool 51
4.4.4 Testing and use 52
4.5 Summary 52

5 Customizing and testing the developed methodology **53**
5.1 Philips Lighting 54
5.1.1 Production process of incandescent lamps 54
5.1.2 Characterization of production process 54
5.1.3 Current process quality control techniques 55
5.2 Monitoring the process 55

5.2.1 Stem making machine 55
5.2.2 Mounting machine 56
5.2.3 Sealing and pumping machine 57
5.2.4 Threading and finishing machine 57
5.3 Scope determination 58
5.4 Analysis technique customization 58
5.4.1 Entitlement calculation 59
5.4.2 Fitting a distribution 62
5.4.3 Entitlement check 67
5.5 Classification technique customization 68
5.5.1 Binary feature selection 68
5.5.2 Training the tool 69
5.6 Testing the customized methodology 73
5.6.1 Classification rate optimization 75
5.7 Summary 76

6 Conclusions and recommendations **79**
6.1 Conclusions 79
6.2 Recommendations 80

Bibliography **83**

A Camera tool **87**

B Training cases **89**

C Matlab m-file of algorithm **91**

D Test cases **95**

E Matlab m-file of alternative algorithms **99**

List of Symbols

Acronyms

ADC	Automatic Defect Classification
ANO	Annealing Oven
ANOVA	Analysis of Variance
CCD	Charge Coupled Device
CF	Cap Filling Machine
DOE	Design of Experiments
FINM	Finishing Machine
LIW	Lead-in-wires
LTM	Lamps Transport Machine
MINQUE	Minimum Norm Quadratic Unbiased Estimator
MLE	Maximum Likelihood Estimator
MM	Mounting Machine
OCMM	Optical Coordinate Measurement Machines
PCA	Principal Component Analysis
PM	Pumping Machine
PQC	Process Quality Control
REML	Restricted Maximum Likelihood Estimator
SM	Sealing Machine
SMM	Stem Making Machine
SPC	Statistical Process Control
SSA	Spatial Signature Analysis
TRM	Threading Machine
WW	WonderWare

Symbols

α	shape parameter of gamma and Weibull distributions
β	scale parameter of gamma and Weibull distributions
$\delta_{new}(x, y)$	feature comparison result
$\delta_{tr}(x, y)$	feature comparison result
λ	shape parameter of Poisson distribution
λ_i	eigenvalues corresponding to normalized eigenvectors
μ	distribution average
$\widehat{\mu}$	estimator of distribution average
μ^T	average tonnage waveform signal
π_{ji}	proportion of cases in class j that contains feature i
ρ	correlation coefficient
σ	distribution standard deviation
$\widehat{\sigma}$	estimator of distribution standard deviation
$\sigma(x, y)$	feature comparison result
a_j	principal components
A	specific characteristic
$\mathbf{A}_k$	quality transformation information matrix at stage k
bf_i	binary feature i
$\mathbf{B}_k$	quality affection information matrix at stage k
cv	coefficient of variation
$\mathbf{C}_k$	measurement matrix at stage k
$d2*$	bias correction factor
$\mathbf{e}_i^T$	experiment vector
$\mathbf{E}$	matrix containing experiments
$f(x)$	mathematical function
m_i	measurement points
n_d	number of datapoints
n_e	number of experiments
n_f	number of features
n_i	number of items
n_{i_t}	number of items for sample of time period t
n_m	number of measurement points
n_s	number of subgroups
n_t	number of time periods
p	chance level
p_A	probability that an item possesses attribute A
p_i	locating points
$\overline{p}_m$	average amount of (semi-)products produced on machine m
p_{m_t}	amount of (semi-)products produced on machine m during period t
$p_{m_{t_{threshold}}}$	deciding threshold value on intention of period

r_{i_t}	counts per unit of area of opportunity of sensor i for period t
r_{i_t}	reject percentage of sensor i for period t (both definitions are equivalent)
r_{ENT_i}	entitlement level for counts per unit of area of opportunity of sensor i
r_{UL_i}	upper limit for counts per unit of area of opportunity of sensor i
$\widehat{r}_{\text{UL}_i}$	estimator of upper limit for counts per unit of area of opportunity of sensor i
$r_{\text{UL}_{i_t}}$	upper limit for counts per unit of area of opportunity of sensor i for period t
rc_i	root-cause class i
$\overline{R}$	average range of the subgroups
R_i	range of the i^{th} subgroup
$R_{i->j}$	two-point moving range between datapoint i and j
s	subgroup size
$\mathbf{s}_t$	system state vector for period t
$\mathbf{s}_{new}$	system state vector of new case
$\mathbf{s}_{tr}$	system state vector of training case
$\mathbf{S}$	sample covariance matrix
$SIM(x, y)$	Similarity between cases x and y
T_m	glass melting temperature
$\mathbf{u}_k$	process fault vector at stage k
v	event
$\mathbf{v}_i$	normalized eigenvector of the sample covariance matrix
$\mathbf{V}$	eigenvector matrix
$\mathbf{w}_k$	noise input vector at stage k
x	count level
$\overline{\overline{x}}$	average of subgroup averages
x_{i_t}	count value of sensor i for time period t
$\overline{x}_i$	subgroup average of the i^{th} subgroup
x_{ENT_i}	entitlement level for counts of sensor i
x_{UL_i}	upper limit for counts of sensor i
$\widehat{x}_{\text{UL}_i}$	estimator of upper limit for counts of sensor i
$\mathbf{x}_k$	state vector at stage k
y	output
$\mathbf{Y}$	experiment deviation matrix
$\mathbf{y}_k$	product quality deviation measurement vector at stage k
$\mathbf{y}_i^T$	experiment deviation vector

Chapter 1

Introduction

In order to improve quality and productivity, it is highly desirable to have a fast, accurate, and robust *process quality control* methodology to identify and eliminate root-causes of quality-related problems for manufacturing processes.

Only in the most advanced and capital intensive industries as the wafer [Tob97, Gle97] and automobile [Ceg96, Apl01] industry, such methodologies have been developed. Current practice in most industries is a product-inspection oriented measurement strategy. Finished and intermediate products are measured and compared with specifications in the product design. If products appear to be outside of the specification limits the causes of this deviation need to be determined. This search is usually based on the experience of operators and sometimes it is very time consuming.

Unfortunately, there are major difficulties in translating the developed process quality control methodologies to other industries. The developed methodologies rely heavily on the specific sensor signals available in each type of industry. Especially the continuous character of the sensor data in the most advanced industries as opposed to discrete data in some of the other industries prevents the use of these techniques.

The term *continuous sensor* is introduced to indicate sensors that measure a continuous quantity and produce a continuous output accordingly. Other industries, that produce cheaper, simpler products that require less accuracy are often predominantly monitored by *binary sensors.* These sensors measure a discrete quantity, or detect an event or presence, and produce a binary output accordingly. Alternatively a continuous quantity is measured but compared to a threshold.

Two issues seem to contribute to a binary world view in these kind of processes. First, because of the less stringent demands on accuracy, it is not necessary in these industries, and therefore too expensive, to control the exact product dimensions or properties (as is common practice in e.g. the semi-conductor and automobile industry). It suffices to produce the product and check for compliance with certain requirements, and accept or reject the product accordingly.

Second, because of their simplicity, these products are often characterized by typically binary features. For example, a beer bottle in the packaging industry either has, or does not have, a crown cork on top of it. The quality of the placement of the crown cork would be difficult and therefore too expensive to measure. Therefore a binary sensor that checks the mere presence suffices. Equivalently, an incandescent lamp in the lighting industry either has, or does not have, a coil connecting the lead-in-wires. Measuring the quality of placement of the coil is very complex and expensive. In this industry it is more interesting to know which part of the produced lamps is compliant. No process quality control methodologies exist however that can indicate real-time whether the resulting level is satisfying and what should be done to improve it.

1.1 Objective

The **aim of this research project** is to develop a *process quality control methodology* that is able to *identify root-causes* of quality-related problems in manufacturing processes that are predominantly monitored with *binary sensors.*

As such this report contributes in expanding the area of process quality control to a multitude of different production processes.

For this development a guiding context is needed as a platform for testing the newly developed tools. The production lines of **Philips Lighting** offer a suitable environment and will form the practical case in this project. This case is used to illustrate how such a methodology can be customized to a specific production process.

The second contribution is therefore the development of a useful quality control tool that has the potential to substantially improve manufacturing performance at Philips Lighting.

1.2 Approach

In Chapter 2 the concept of process quality control is discussed. Next, a framework is built for comparison of existing quality control methodologies. Furthermore, a categorization of production processes is presented that allows to evaluate the compliance of a specific process with each methodology.

Using this framework, Chapter 3 then explores literature and contemporary research on quality control in the most advanced industries, in which continuous sensors are used for monitoring. The *monitoring*, *analysis* and *classification* of quality-related problems along with different appropriate ways of modeling production processes will be discussed.

The framework will serve as a guidance in the stepwise development of the binary sensor - methodology. Chapter 4 evaluates the applicability of the techniques presented in Chapter 3 and combines, adapts and extends these techniques yielding a quality control methodology for binary industries[1].

Chapter 5 then illustrates how the developed methodology can be made operational for the production lines of Philips Lighting. The methodology is tested and its performance is evaluated.

Finally, the conclusions and recommendations for further development are presented in Chapter 6.

[1]this term is introduced to refer to industries in which predominantly binary sensors are used

Chapter 2

Process Quality Control

Process quality control consists out of a series of iterative steps with the objective of preserving process control. First, these steps will be explained in detail.

The applicability of a certain methodology to a specific production process is dependent on the ability of the process to deliver the needed information. Therefore, a categorization of the possible types of production processes is discussed next. This categorization facilitates the presentation of an insightful overview of existing methodologies in Chapter 3.

2.1 Process Quality Control loop

Process quality control aims at maintaining optimal process conditions [Ott00].

process quality control - a set of tools that is employed to maintain *that* quality level of the production process in which only common causes are present

With common causes those causes are meant that always tend to be present to some extent and are nearly indistinguishable from each other.

The starting point is a process that is in a state of control. Logically, it needs to be monitored whether this state changes. If so, appropriate action needs to be taken.

Several distinct steps can be distinguished as illustrated in Figure 2.1.

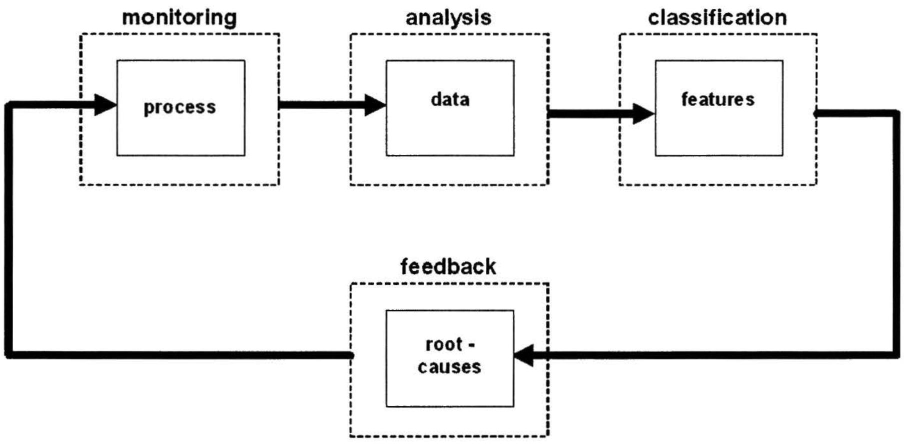

Figure 2.1: Process Quality Control – loop

The generally accepted view is that product quality is achieved by controlling process quality [Hop01]. Therefore, quality control starts with monitoring the production **process**. Manual and automated inspection is performed on *product* and *process* parameters that are in some way indicative of the process its quality level.

In the next step, the generated quality-related **data** is analyzed, yielding information about the state of the process.

The conclusions of the extracted information are presented as certain **features** of the system state. These features can take many different forms. Depending on the specific analysis techniques, these features may be as specific as exact process faults or as broad as any derived variable that contains some quality information.

The features are the input for the classification of the system state into distinct root-cause classes. As such the present **root-causes** of problems are retrieved. The root-causes are then finally used as input for feedback into the process. The right feedback should reestablish process control.

Automated monitoring of a process is performed by sensors. Many different types exist, a brief overview is presented in Section 3.1. As stated earlier, an important distinction can be made between sensors that yield a continuous output and those that perform a binary check. The term *continuous monitoring* is introduced for monitoring with the former type of sensors and *binary monitoring* for predominant use of the latter type (see Figure 2.2).

If relevant direct and continuous data on quality parameters has been retrieved the information density can be very high. This greatly enhances the possibility for successful data analysis to yield informative features of the system state. If these features are clearly known, the subsequent classification is simplified as well.

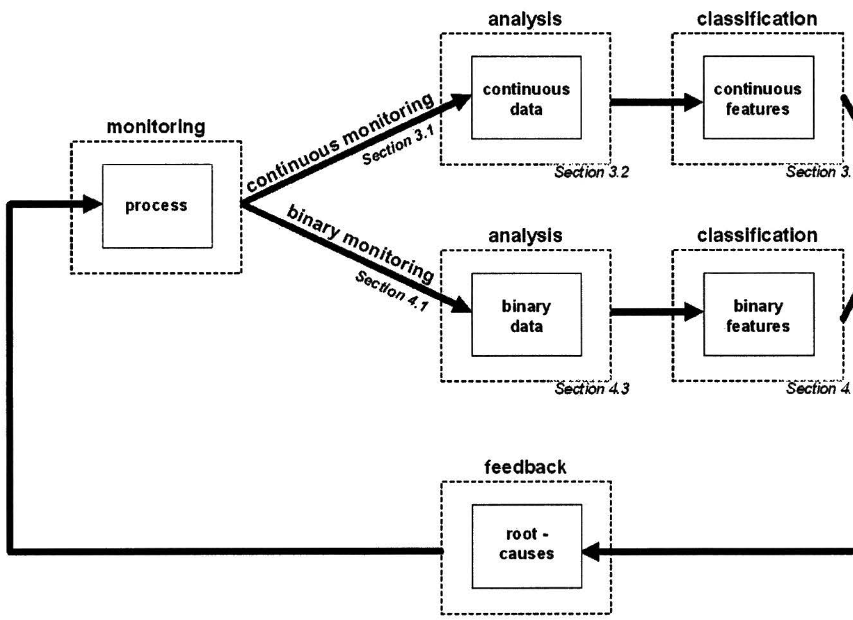

Figure 2.2: PQC – loop indicating the differences caused by continuous and binary monitoring

Binary data, on the other hand, typically offers an indirect and therefore less informative look at the quality of the process. This poses a huge challenge on the successful continuation of the quality control loop.

The upper branch in Figure 2.2 represents the analysis and classification techniques that have been developed for processes that are monitored with continuous sensors. Some interesting and insightful examples will be presented in Chapter 3. Extensive literature review did not reveal any methodologies readily applicable to the lower branch in the figure. It is the aim of this report to develop a combined analysis and classification tool that is capable of retrieving the root-causes of present problems for processes that are monitored by binary sensors. The subsequent feedback step will not be considered.

2.2 Categorization by production process

Production processes can be categorized in several ways. Important factors from the quality control – point of view are the *nature* and *structure* of the production process and the *costs* associated with producing nonconforming products ([Gro96],[Hop01]).

A classification of production processes by *nature* focusses on the type of operations involved in the manufacturing process. This determines which quality parameters can be checked and therefore the type of sensing techniques (and thus sensors) that can be used. Figure 2.3 presents such a categorization.

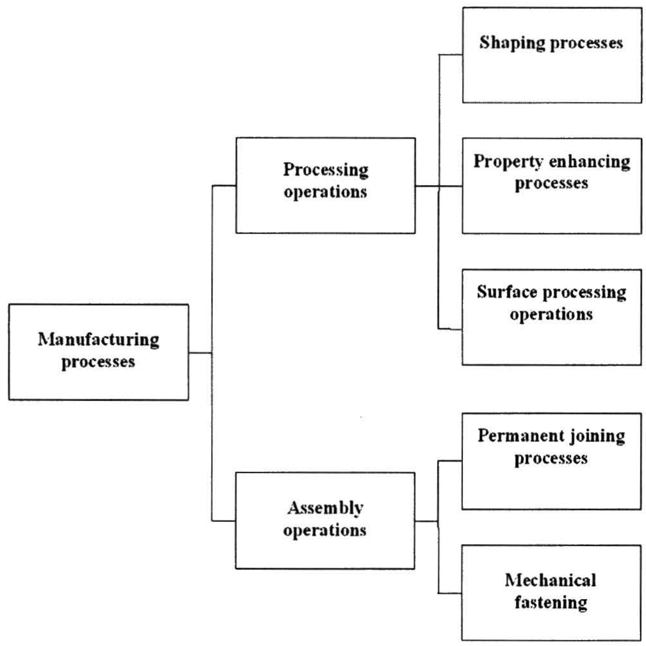

Figure 2.3: Categorization of production processes by **nature** (adapted from [Gro96])

It is very well possible that a production process involves (many) different operations, and therefore possesses a mixed nature. Normally, the process can then be divided into a number of parts with a different operation being performed at each part. Therefore, ultimately each part can be characterized by a different process nature.

The process *structure* is defined as the manner in which material moves through the plant [Hop01]. The chosen structure for a certain production process typically depends on the volume and the level of standardization of the process. Higher volumes and standardization heavily increase the efficiency of automation. The most common categorization by process structure is presented in Figure 2.4.

The third relevant aspect of production processes is formed by the *costs* that are associated with producing nonconforming products. If these costs are relatively high, large expenditure on research and equipment for process control automation is justified. Automation of process control is desirable because it can achieve performance beyond the limits of human cognitive capability and will ultimately reduce man cost. The large expenditure that is usually involved in the research and equipment for such automation needs to be outweighed by the savings resulting from the decrease of nonconforming products.

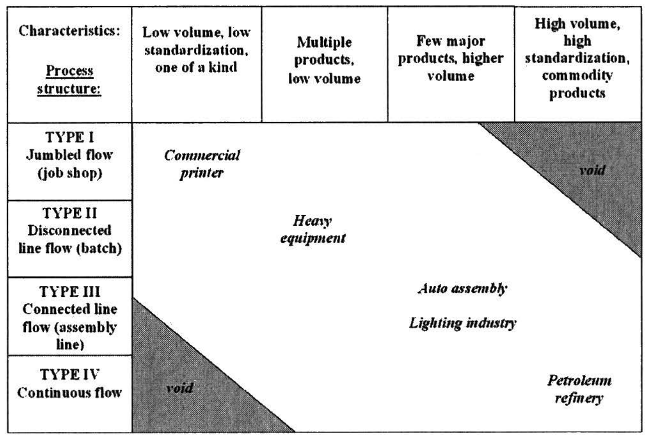

Figure 2.4: Categorization of production processes by **structure** (adapted from [Hop01])

The *costs* associated with producing nonconforming products are therefore very influential. In this respect, these costs should be interpreted very broadly, including their:

- Material costs
- Production costs
- Costs of rework
- Possible harm to people
- Possible harm to the environment
- Possible failure to comply with production orders

High levels of automation and expensive sensing solutions are easily justified if they are outweighed by these opportunity costs.

2.3 Summary

In this Chapter the concept of process quality control has been introduced. The distinct steps are formed by the monitoring of the process, the analysis of the data, the classification of the system state based on the retrieved features and the feed-back based on the found root-cause. This framework allows a step-wise discussion of existing continuous methodologies as well as a step-wise development of a binary one.

Furthermore, a categorization of production processes by their nature, structure and costs of producing nonconforming products was presented. This categorization facilitates the overview of the techniques that will be reviewed.

Chapter 3

Review of continuous methodologies

In this Chapter various continuous process quality control methodologies are discussed. First, attention is devoted to the sensing devices used for monitoring. Next, analysis techniques aimed at extracting information from the monitoring step are discussed. Finally, tools will be discussed that are used to classify system states based on this information yielding the root-causes of the problems that are present.

The obtained overview will serve as a reference for the *step-wise* development of the binary methodology in Chapter 4.

3.1 Monitoring techniques

A multitude of different sensing techniques exists. They are based on the detecting and transducing capabilities of several distinct measuring elements. An overview of some of these measuring elements is presented in tables 3.1 and 3.2.

The sensing techniques that are used in a particular situation, together with the level of automation, characterize the automated monitoring process. The *applicability* of these techniques depends on the nature (and sometimes on the structure) of the production process. Their *suitability* is determined by the process structure and the costs associated with producing nonconforming products.

Mechanical measuring element	**Some examples**
Contacting spindle	-
Elastic member	Load cells Proving ring, Diaphragm
Mass	Seismic mass, Pendulum
Thermal	Thermocouple, Bimaterial, Chemical phase
Hydropneumatic	Hydrometer (static), Venturi (dynamic)

Table 3.1: Overview of some sensing techniques based on mechanical measuring elements (adapted from [Bec95])

Electrical measuring element	**Some examples**
Resistive	Contacting, Variable-area conductor
Inductive	Variable coil dimensions, Moving coil
Capacitive	Changing air gap
Piezoelectric	-
Semiconductor junction	Junction threshold voltage, Photodiode current
Photoelectric	Photovoltaic, Photoemissive
Hall effect	-

Table 3.2: Overview of some sensing techniques based on electrical measuring elements (adapted from [Bec95])

3.1.1 Effectiveness of monitoring

The effectiveness of a sensor system is characterized by the diagnosability that is offered by the system, which is defined as its capability to identify major variation sources [Din03a]. The distribution of the sensors in the production system plays a key role in this respect. Traditionally, existing literature on this topic has a twofold focus:

- optimization of *multi-station* sensor allocation for the purpose of product inspection (e.g. [Lin64, Yum81])
- optimization of allocation of sensors for the purpose of variation diagnosis at a *single* measurement station (e.g. [Udw94, Kha99])

More recently, the integration of sensing information from different measurement stations into a single state-space model has been investigated [Din03b], which yielded the optimal allocation of sensors across (and inside) stations via the mechanism of variation propagation.

3.2 Analysis techniques

3.2.1 State Space Modeling

An important class of analysis techniques is based on *state space modeling*. It is most commonly used for processes that posses a predominant assembly nature, possibly combined with some shaping and other processing operations (see categorization of Figure 2.3). This type of modeling is especially useful in multi-stage manufacturing processes, that possess a connected line flow structure (type III of categorization of Figure 2.4) [Ceg96]. Important examples of such manufacturing processes can be found in the automobile industry and in metal cutting industries [Zho03].

The general state space modeling approach starts with the development of a *variation propagation model* which describes how dimensional deviations of the product propagate from one process stage to the next. This model can be based on design information of the product and the process (as in [Hua01]), for example on CAD information about the fixture geometry and location of the measurement points [Ceg96] as is shown schematically in Figure 3.1.

The measurements in the automobile industry are normally performed with Optical Coordinate Measurement Machines (OCMM) that are installed at multiple locations in the multistage manufacturing process. These OCMM's work with CCD cameras and are based on the photo-voltaic sensing technique from table 3.2.

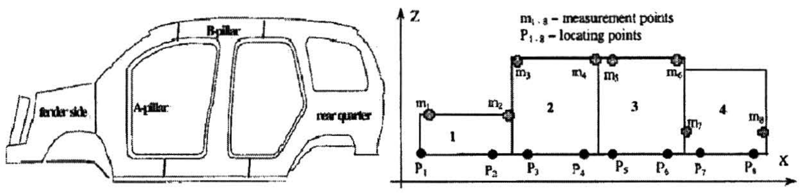

Figure 3.1: Fixture geometry (P_i) and location of measurement points (m_i) in an automobile manufacturing process (adapted from [Din02])

In connected line flow (type III) processes, variation in the product is accumulated as the product moves from stage to stage in the production line, as illustrated in Figure 3.2 [Zho00].

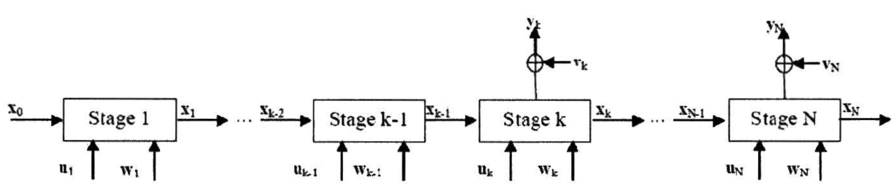

Figure 3.2: Variation propagation in multi-stage manufacturing process (taken from [Zho00])

The product quality information (i.e. the dimensional deviation) at stage k is represented by the state vector $\mathbf{x}_k$. Vector $\mathbf{u}_k$ describes the process faults. Fixture errors, machining errors, thermal errors, and so on, manifest themselves as shifts in the *mean value* or increases in the *variance* of this vector. Un-modeled errors are represented by the noise input vector $\mathbf{w}_k$. The measurements of product quality deviation are denoted by $\mathbf{y}_k$. Finally, the measurement noise is denoted by a zero mean random vector $\mathbf{v}_k$.

A linear state space model can then be built to describe the product quality information flow, yielding (see [Zho00] for the complete derivation):

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{3.1}$$

In which $\mathbf{A}_{k-1}\mathbf{x}_{k-1}$ represents the quality information transformation from stage *k-1* to stage k. $\mathbf{B}_k\mathbf{u}_k$ describes how the part quality is affected by the process faults, and $\mathbf{C}_k$ is the measurement matrix that maps the product quality characteristics to the measurements. The system matrices $\mathbf{A}_k, \mathbf{B}_k$ and $\mathbf{C}_k$ describe the interaction information between the process and the product.

The state space model can simply be constructed by examining the relation between $\mathbf{x}_{k-1}$, $\mathbf{x}_k$ and $\mathbf{u}_k$ locally, at each stage k. The large body of single-stage operation – literature can therefore be used.

$\mathbf{y}$ is the deliverable of the monitoring phase, so after building the model it needs to be solved for the process faults $\mathbf{u}$ with the use of estimation techniques for linear mixed models.

Typical statistical estimation algorithms are ANOVA, Maximum Likelihood Estimation (MLE), Restricted Maximum Likelihood Estimation (REML) and Minimum Norm Quadratic Unbiased Estimation (MINQUE). A comparison of these methods can be found in [Din03b], a review in [Rao98].

3.2.2 Spatial Signature Analysis

The semiconductor industry provides an important example of manufacturing processes that are characterized by surface processing operations (see categorization of Figure 2.3). The primary monitoring tool for the investigation of wafer defects is optical inspection [Tob97]. High-resolution images of individual defects are collected off-line to assess problems in the manufacturing process. Low-resolution defect wafermaps[1] from in-line optical inspection tools are used as a less time consuming and less expensive alternative.

Trends towards larger semiconductor wafer formats and smaller critical dimensions have led to an exponential increase in the volume of visual and parametric defect data however. Consequently, automation of analysis has become a necessity [Tob97]. Automation of defect analysis on an optical basis can be performed for jumbled, disconnected and connected flow processes (see categorization of Figure 2.4).

A promising emerging technology that is based on artificial intelligence is Spatial Signature Analysis (SSA). A *spatial signature* (see Figure 3.3) is defined as a unique distribution of wafer defects originating from a single manufacturing problem.

The method tries to capture operator experience with respect to wafer defect analysis through a teaching method. The core is formed by an image processing, fuzzy classifier system that is able to distinguish between different types of quality defects [Tob97]. In this way the bulk of product quality data is transformed into the assignment of a defect wafer into a single *elemental set*.

The first step in the Spatial Signature Analysis is the conversion of the electronic wafermap file into a gray-scale image. Each pixel in this image is assigned an intensity value according to the number of defects in the corresponding area. As such, each *pixel* groups the individual defects on a first level. The connection of pixels to each other, based on their proximity, then forms *clusters* of defects. A clustering procedure has been developed that groups clusters of pixels into multi-element *objects* (such as

[1] a wafermap is a list of defect coordinate locations generated by an optical- or laser-based wafer inspection tool [Gle97]

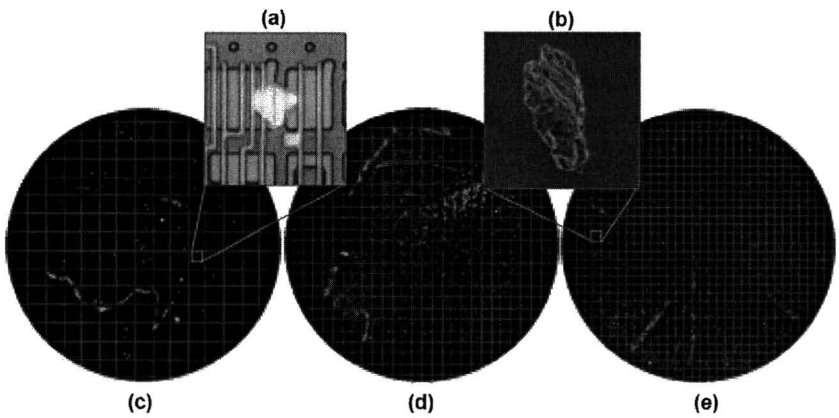

Figure 3.3: Spatial Signatures (taken from [Tob97]) (a) High-resolution optical defect image (b) High-resolution Scanning Electron Microscope defect image (c) Single wafer containing scratch signatures (d) Stack of wafers superimposed highlighting a subtle systematic particle contamination problem (e) Single wafer showing a spin-coater streak pattern

scratches or streaks). Finally, objects are grouped into *elemental sets*, depending on their morphology and their proximity to neighboring clusters [Tob97].

The different elemental sets are global, curvilinear, amorphous and micro-structure. Sparsely distributed objects (such as a ring pattern or a random uniform distribution of particles) are assigned into the *global* set. Elongated objects such as scratches or streaks belong to the *curvilinear* set based on attributes as elongation, compactness and orientation. The *amorphous* set contains tightly clustered objects. The *micro-structure* set, finally, contains distributions of pixels whose sub-pixel defects are organized in a linear fashion [Tob97].

The resulting input into the classification step thus consists of an assignment of the wafer into an elemental set, and a characterization of its features. On this knowledge, fuzzy techniques are employed to classify the wafer in specified root-cause classes.

An alternative to SSA is Automatic Defect Classification (ADC). Most ADC systems use reference-based image analysis. A segmentation algorithm localizes the defect by comparing a defect image with a defect-free reference image. A defect mask is generated by the algorithm that describes the location and extent of the defect. Features are then extracted from the mask and the original pair of images that uniquely describe the appearance of the defect. A defect classification algorithm then attempts to automatically categorize the new defect based on training exemplars that are provided by the expert human classifier [Gle97].

3.2.3 Diagnostic Feature Extraction with the use of Principal Component Analysis

Yet another type of manufacturing processes is formed by the shaping processes (see Figure 2.3). Stamping processes (see Figure 3.4) are an example of this category.

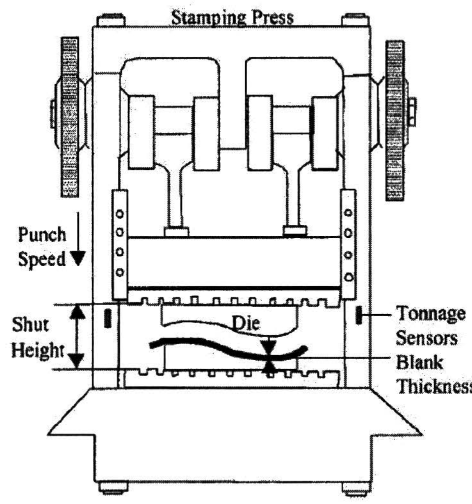

Figure 3.4: Example of a shaping process (taken from [Jin00])

In the monitoring phase of these processes, the stamping force is measured by strain gauges (the electrical - resistive type of table 3.2). The resulting tonnage signal can be analyzed to provide information about changes in the stamping process that may harm product quality [Jin00].

Due to the complexity of a stamping process, many process variables can influence tonnage signals. In practice, the interactions among the stamping process variables are very significant and complex. Recent multivariate approaches [Jin99] take these correlations into account. In this way analysis tools have been developed that indicate whether a process is normal or abnormal, by detection of multivariate waveform changes.

It is difficult to deliver the occurring features in the right form to the classification tool for root-cause finding, however. Several publications ([Rob95], [Koh95], [Che97]) have addressed this issue, while focussing on single-fault situations without consideration of the interactions among the process variables. These detection criteria are effective only when it is realistic to assume that all other variables are unchanged, which is normally not the case [Jin00].

A much more interesting analysis approach is the preparation for the classification step that consists of the extraction of diagnostic features. With the use of the fractional factorial design of experiments (DOE), [Jin00] proposes a new diagnostic feature-extraction methodology. An overview of the analysis steps is presented in Figure 3.5.

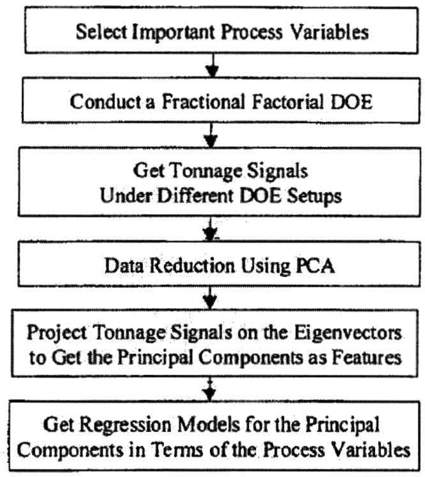

Figure 3.5: Diagnostic Feature Extraction approach (taken from [Jin00])

If too many variables influence the process the testing of these variables may become too complex. It is therefore necessary to select the most relevant ones, based on process engineering knowledge. If there is not sufficient initial knowledge, a screening experiment is suggested by [Jin00]. In the stamping process case, variables as material thickness, shut height and punch speed have been selected (see Figure 3.4).

The next step is to conduct the fractional factorial design of experiments to study the effect of the selected process variables. In the DOE each process variable is set to normal or abnormal[2]. For every possible combination of the process variables (in terms of normal versus abnormal), the fault patterns of tonnage signals are then obtained (with the use of the strain gauges). Apart from the main effect of each variable, the interactions between two variables are of interest as well.

With the use of Principal Component Analysis (PCA) techniques the resulting vast amount of highly correlated data is reduced to a lower dimensionality via orthogonal projection [Jin00]. This is achieved via selection of the most significant eigenvectors in the original data.

A tonnage waveform signal obtained from the DOE can be represented by n_m measurement points taken along the process cycle. All n_e experiments can then be put into a matrix $\mathbf{E} \in \mathbb{R}^{n_e \times n_m}$. A row vector $\mathbf{e}_i^T$ of $\mathbf{E}$ then represents one cycle.

[2]values corresponding to normal and abnormal situations are again based on process engineering knowledge

In order to obtain a data matrix with deviations caused by changed process variables the average tonnage waveform signal μ^T is needed. μ^T can be estimated by:

$$\overline{\mathbf{e}}^T = 1/n_e \sum_{i=1}^{n_e} \mathbf{e}_i^T \tag{3.2}$$

The row vectors of $\mathbf{Y}$ are then obtained by subtracting this average from each signal:

$$\mathbf{y}_i^T = \mathbf{e}_i^T - \overline{\mathbf{e}}^T \tag{3.3}$$

The resulting matrix can then be decomposed with the help of PCA [Joh98] as:

$$\mathbf{Y} = a_1\mathbf{v}_1^T + a_2\mathbf{v}_2^T + ... + a_p\mathbf{v}_p^T \tag{3.4}$$

The vector $\mathbf{v}_i \in \mathbb{R}^{n_m \times 1}$ $(i = 1, ..., n_m)$ is the normalized eigenvector of the sample covariance matrix $\mathbf{S}$ of $\mathbf{Y}$ according to:

$$\mathbf{S}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{3.5}$$

$$\mathbf{S} = \frac{1}{n_e - 1} \sum_{i=1}^{n_e} \mathbf{y}_i \mathbf{y}_i^T \tag{3.6}$$

where λ_i is the eigenvalue corresponding to the eigenvector $\mathbf{v}_i$. $\mathbf{V}$ is the matrix consisting of the eigenvectors $\mathbf{v}_i$ $(i = 1, ..., n_m)$ and forms an orthogonal basis for the space spanned by $\mathbf{Y}$. The principal components $a_j \in \mathbb{R}^{n_e \times 1}$ are finally obtained by projecting $\mathbf{Y}$ onto the vector $\mathbf{v}_j$ $(j = 1, ..., n_m)$ as [Jin00]:

$$a_j = \mathbf{Y}\mathbf{v}_j \tag{3.7}$$

It is often observed that when a data set is projected to eigenvectors, typically a few eigenvectors that correspond to larger eigenvalues are associated with the systematic process variations, whereas the remaining eigenvectors tend to reflect the variations of the process noise [Joh98]. This first group of eigenvectors then defines the lower-dimensional orthogonal space to which the original data is projected. In this way the dimension of the original PCA model, n_m, is reduced to a lower dimension n_f. The original large dimension of the tonnage signal $\mathbf{e}_i^T = [e_{i1}, ..., e_{in_m}]$ is therefore expressed by a smaller dimension of *features* $[a_{i1}, ..., a_{in_f}]$.

The effect of the process variable changes (caused by certain root-causes) on the tonnage waveform signals will be represented as the changes of these *features*, which were explained to be the most significant *principal components*. These features are therefore the deliverable of the analysis tool.

3.2.4 Statistical Process Control

Statistical process control is a tool that is able to distinguish random behavior from non-random behavior. Monitoring data of any process quality indicating parameter belonging to conforming products, or to fault-free production runs, is retrieved by (normally continuous) sensors [Whe95]. The measurements taken from a stream of products, or the measurements of a process parameter retrieved at a certain frequency, can be presented as: x_1, x_2, x_3, ..., x_n.

If these measurements are grouped into histograms, for different production runs, two results are possible. The periodic histograms of the measurements either show a consistent pattern of variation, or they will vary from run to run. In the first case, the output distribution does not change over time and the process is therefore said to be in a state of statistical control. If it does change however, it is said to be out of control [McC98].

When a process is in control some variation will still be present. This variation is due to common causes however, which are defined as [Whe95]:

common causes - those causes of problems that are attributable to the design of the process and therefore affect all outputs of the process.

These common causes can be categorized into the *methods*, *materials*, *machines*, *personnel* and *environment* that make up a process and the *inputs* required by the process [McC98]. The individual variations may be thought of as being created by a constant system of a large number of chance causes in which no cause produces a predominating effect [She80]. It is therefore not profitable to try to determine and remove the causes of the uncontrolled variation [Whe95].

If a process is out of control there are some special causes of variation, termed the assignable causes [Whe95]:

assignable causes - events or actions that are not part of the process design. They are typically transient, fleeting events that affect only local areas or operations within the process (such as a single worker, machine part, or batch of raw materials or components) for a brief period of time.

In this situation it will be profitable to try to determine and remove the cause of the uncontrolled variation, which is achieved by diagnosing the problem and performing relevant feedback.

It is useful to approximate the histogram of the stable pattern of variation by an arbitrary (and usually continuous) mathematical function, $f(x)$. This function is called the probability distribution function (or probability density function).

It is defined so that the product $f(x)dx$ approximates the proportion of measurements that fall between the values of x and $x + dx$ [Whe95]. $f(x)$ can be characterized by a measure of location (such as the arithmetic average (μ) or the median) and a measure of dispersion (such as the range, the variance (σ^2) or the sample standard deviation (s)). Normally the average μ and the standard deviation σ or their estimators $\overline{x}$ and s are taken [McC98].

Standard SPC monitoring techniques divide the possible values of a datapoint, into 6 zones, each with a width of σ, covering the range between $\mu - 3 \cdot \sigma$ and $\mu + 3 \cdot \sigma$. A typical division is shown in the following $\overline{x}$-chart (Figure 3.6).

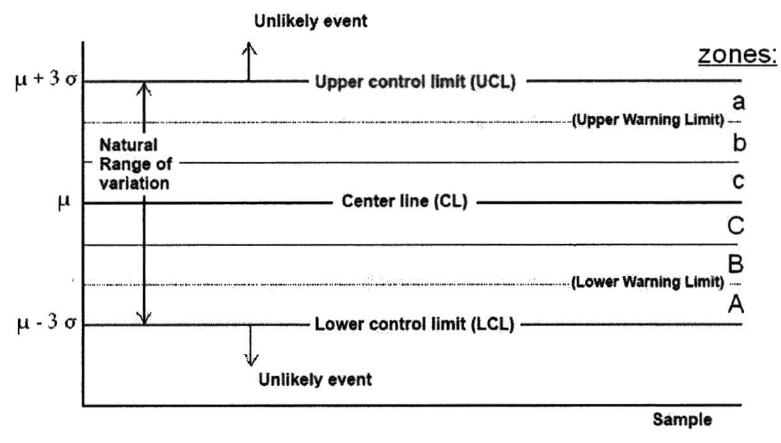

Figure 3.6: $\overline{x}$-chart with SPC control limits and zones

If datapoints fall below *zone A* or above *zone a*, the chance of this being a random fluctuation can be calculated. Assuming the process is in statistical control and therefore justly described by a certain probability distribution function, this chance is known. In the case of a normal distribution, for example, this chance is only 0.03 %. So, in this case, with 99.97% certainty it can be said that abnormal behavior has occurred during the time interval that is the basis of this datapoint. The conclusion therefore is that the process has been out of control during that period.

There are also other checks [Whe95](all with at least 99.5% certainty) that indicate that the process is *out of control*, as summarized below:

- point beyond control limits
- run of 9 points above or below center line
- 4 out of 5 points in zone b/B or beyond
- 2 out of 3 points in zone a/A or beyond
- trend of 6 points going up or down
- cumulative sum of deviations above or below a certain limit

Apart from this $\overline{x}$-chart that monitors the *mean* of the process, a chart can be created for monitoring of the *variance*. This chart, the R-chart, is created in much the same way as the $\overline{x}$-chart. The checks for abnormalities in the variance of the process also coincide.

Tampering with processes that are in control may very well lead to worsening of the situation [McC98]. Therefore, only if it is almost certain that the process is out of control it is sensible to diagnose it to find the *assignable cause* of this special variation. Only in this situation, feedback based on the root-causes that are found is sensible.

3.3 Classification techniques

As illustrated in Figure 2.1, the aim of the classification step is to indicate the root-causes of quality related problems that are present in the production process. Hereto, classification techniques use information that comes from the analysis step. The analysis techniques of the previous Section presented some examples of data processing that lead to informative features.

The objective of this Section is to present an overview of some important classification techniques. Hereto the techniques are qualified as direct, semi-direct and indirect, referring to whether a technique pinpoints root-causes directly, via experiments, or via comparison to training cases. To be able to indicate root-causes most of the techniques need to be trained on which root-causes belong to which features.

3.3.1 Direct techniques

The state space modeling technique (of Section 3.2.1) is an example of an analysis tool that models the link between root-causes of problems and features. Therefore, when the present features are known (presented as process faults $\mathbf{u}$), the case can directly be classified in a root-cause class.

As explained, after measuring the resulting part quality $\mathbf{y}$ in the monitoring step, the set of equations 3.1 can be solved for $\mathbf{u}$ ([Zho00]) with statistical estimation algorithms as ANOVA or MLE (see [Rao98]). The link between root-causes of problems like specific fixture errors and features (i.e. dimensional deviations) is modeled as $\mathbf{B}_k$. The process fault vector $\mathbf{u}_k$ at stage k therefore essentially contains the present root-causes already, represented as shifts in the mean and variance of its elements.

Statistical hypothesis testing (see e.g. [Mon99]) is used to assess the statistical significance of observed deviations in this vector, and thus the certainty level of presence of the root-cause.

3.3.2 Semi-direct techniques

In the diagnostic feature extraction technique experiments are executed to observe the effect of changes in process variables on certain features. Hereto, the obtained *features* are used as the response variables in a DOE regression analysis. Via Analysis of Variances (ANOVA) it is investigated whether a *process variable* has a significant effect on the selected *features* for the different experiments. The significant variables that have the major contributions to a feature's variability are called its *diagnostic variables.* These diagnostic variables are the *root-causes* behind the occurrence of the features.

The knowledge obtained via the experiments can therefore be used to classify new cases, based on their observed features, to root-cause classes.

3.3.3 Indirect techniques

Indirect techniques make use of the observed link in training cases between features, as delivered by an analysis tool, and occurring root-causes.

Different indirect classification techniques can be distinguished:

Decision boundary

An example of a technique that can be directly applied to monitoring data is linear modeling. A mathematical relationship that characterizes the sensor values for situations with a certain root-cause is retrieved by using the least squares method [Bis03]. In the same way the situation without the root-cause is modeled.

In Figure 3.7 an example is illustrated in which a linear decision boundary is fitted in between these mathematical expressions. This decision boundary is then used to estimate the presence of the root-cause for new cases. The input variables from the monitoring step are x_1 and x_2. In this example the linear model is able to provide a good estimate of the presence based on this input.

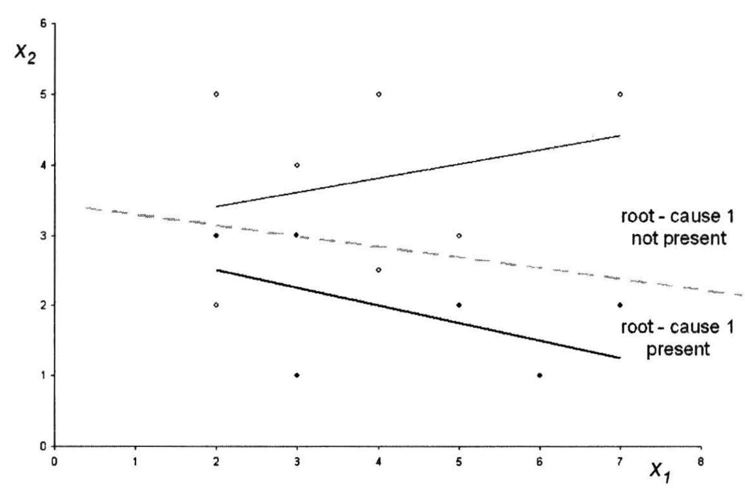

Figure 3.7: Linear decision boundary for root-cause presence. x_1 and x_2 are sensor values or features. Black dots indicate cases in which root-cause 1 is present, whereas white dots indicate cases without the root-cause.

In practice, more input variables lead to a higher dimensionality of the domain. A more complex relationship between input and root-causes furthermore leads to a higher dimensionality of the decision boundary. In finding a decision boundary that is a good approximation of reality a trade-off exists between *bias* and *variance*, caused by lower and higher order approximations respectively [Bis03]. The bias is a result of the inflexibility of lower order polynomials for modeling the boundary, whereas the *variance* results from overfitting by higher order polynomials.

Learning algorithms

Learning algorithms aim at classifying new cases to classes, dependent on some features. These features can be delivered by data pretreatment techniques of the analysis step. Essentially these algorithms calculate a similarity between a new case and each of the training cases, based on a feature-by-feature comparison [Win98]. An example of a learning algorithm is Automatic Defect Classification (ADC), which was presented as an alternative approach to the spatial signature analysis discussed in Section 3.2.2.

The backbone of a learning algorithm consists of a similar principle as visualized for the decision boundary approach in Figure 3.7. The difference lies in the capability of the algorithm to weigh deviations in different directions in the multi-dimensional space in a different manner. This is an advantage over least-square methods. Furthermore, no real decision boundary is fitted, but conceptually a virtual one is created as cases are classified in the described imaginary space.

Neural networks

Neural networks are computer algorithms that possess the ability to learn a specific knowledge. They can adapt this knowledge to new situations and are able to provide reliable classifications of data [Pac04]. This specific knowledge is learned by iterating through a set of exemplar data. Two approaches exist. Learning takes place through internal clustering (self-organizing or competitive learning) or through paired training sets (supervised learning) [Hay99].

Advantages of neural networks include their ability to use various types of data, such as multiple input-output data, a combination of qualitative and quantitative data and nonlinear data. A disadvantage however is that it is very difficult to get the theoretical expression for the relation between input and output, which makes neural networks essentially black boxes [Min05].

Fuzzy logic

The unique aspect of fuzzy logic techniques is that they are able to simultaneously handle both objective numerical data and subjective linguistic knowledge [Men95]. Using traditional mathematics it is usually impossible to take the linguistic knowledge, which can be very informative, into account. An example of a fuzzy logic application was found in the spatial signature analysis (see Section 3.2.2).

Another example of an interesting application of fuzzy logic for fault-type classification is described by [Das06]. Here, process faults in power transmission systems are classified based on the differences in sequence and magnitude of the fault current.

Fuzzy techniques consist out of four components: a fuzzifier, rules, an inference engine and the defuzzifier [Men95]. The fuzzifier maps the input numbers $\mathbf{x}$ into fuzzy sets. In this way, rules that are stated in terms of linguistic variables can be applied to the input. The inference engine then handles the way in which the different rules are combined. In many applications the defuzzifier then maps the fuzzy output sets back into numbers ($\mathbf{y}$). As such a quantitative, nonlinear, mapping $\mathbf{y} = f(\mathbf{x})$ between input and output is obtained [Men95].

3.4 Summary

In this Chapter various continuous techniques have been reviewed as an important reference for the exploratory development of a binary technique. First, some sensing principles used in continuous monitoring were introduced. Next, elaborate attention was devoted to insightful analysis techniques. Both techniques that are based on product parameters and techniques based on process parameters were discussed.

Finally, some classification techniques were presented. It was illustrated that classification can either be direct (i.e. based on prior modeling), semi-direct (i.e. based on relationships observed in experiments) or indirect (i.e. based on comparison to diagnosed training cases). During the development of the binary process quality control methodology in the next Chapter, parts of the various techniques will be borrowed, adapted or extended.

Chapter 4

Development of binary methodology

As discussed in Chapter 1, many production processes are predominantly monitored with binary sensors. In general, these production processes produce relatively simple products that require lower accuracy levels and have lower costs associated with producing nonconforming products than in the most advanced industries. These processes are monitored by binary sensors because the simpler products are often characterized by typically binary features. It is furthermore not necessary to control the exact product dimensions or properties during production. It suffices to produce the product and check for compliance with certain requirements, which is often a much cheaper approach.

It would still be very beneficial for these industries to understand when variation in product quality occurs, what causes this variation, and how the optimal process conditions can be restored. As discussed in Section 2.1, the first question is answered in the analysis step, the second in the classification step and the third in the feedback step[1].

The current challenge is to use the obtained view on process quality control to develop a suitable methodology for these kind of processes. This methodology needs to be capable of maintaining process control in environments in which mainly binary data comes out of the monitoring tool. Section 4.1 discusses the characteristics of this type of data.

Next, the applicability and suitability of the presented continuous tools for the binary case is evaluated. Based on this evaluation, Section 4.3 discusses the needed adaptations and extensions to develop a binary analysis technique. In Section 4.4 the subsequent classification step is developed.

[1]As stated before, this latter question will not be discussed in this report

In order to allow application to a multitude of different industries, the current development will take place on as high a level of abstraction as possible. Making the methodology operational for a specific production process then requires additional customization. An exemplary procedure is described in Chapter 5. As the performance of the methodology can only be tested on a practical platform, the performance evaluation is postponed to this section as well.

4.1 Monitoring with binary sensors

The term binary sensor was introduced to indicate a sensing device that produces a binary output. It is therefore only capable of producing a 0 or a 1. Like with continuous sensors, either a **product** or a **process** characteristic can be measured.

In the case of **product** measurements the binary output means that semi-product x *does not*, versus *does*, possess characteristic A. The conclusion of failure to possess this characteristic can already be attached to the check. This would lead to the meaning of *conforming* versus *nonconforming*, or equivalently, *accept* versus *reject* the product. Characteristic A can be many different things, ranging from the presence of a component to a continuous quality related variable like resistance, distance or temperature that is compared to a standard.

If **process** parameters are measured in a binary way, the output possesses the same form, but then in relation to the process. Some characteristics are typically binary by nature, like a certain operating mode being switched on or off. The control system of a process can furthermore be informed on whether a certain process parameter has exceeded a critical value or not, instead of being informed on the precise value.

In transforming the continuous quality related parameter into a binary check, obviously a lot of information on the parameter is lost. Dealing with binary information therefore reduces the opportunities for problem analysis.

So why are not all processes monitored by continuous sensors then? Several reasons exist. First, many parameters like compliance, presence, type, or mode, are binary in the first place. Another important reason for not monitoring the continuous quantity might be the involved complexity and therefore the associated costs. Furthermore, in some cases the lost information would not be helpful in determining the presence or causes of problems.

The consequence is that many production processes exist in which we need to do with the less informative binary data that is delivered by binary sensors.

4.2 Applicability of continuous techniques

4.2.1 Analysis techniques

The continuous analysis techniques get their data from a monitoring system with continuous sensors. These sensors are able to provide real-time values of direct quality related product or process parameters.

Two examples of techniques that monitored **product** parameters have been given. In the *state space modeling* approach (see Section 3.2.1) the dimensions of the semi-product are exactly known throughout the process. The *spatial signature analysis* techniques of Section 3.2.2 demonstrated the use of visual images containing much information on contamination and damage of wafers.

Exemplary for the continuous techniques that monitor direct **process** parameters, Section 3.2.3 presented *diagnostic feature extraction.* In this analysis technique the primary involved process parameter, the stamping force, was monitored for abnormalities.

In **state space modeling**, it is crucial to know the exact dimensional deviations of the components from each production step to the next, to model the variation propagation. Such a complete overview of the dimensions of the semi-product at every stage can not be produced by binary sensors however. Installing continuous sensors of the discussed type at each production step, is likely to be an expensive solution, that is only outweighed by the savings if costs of nonconforming products are very high. As discussed in Chapter 1 this is usually not the case in processes that are monitored in a binary way. Therefore it is not possible to model the link between root-causes and features for the binary case. The **direct classification** techniques that rely on the availability of such a model can therefore not be used. The state space modeling approach does offer the interesting and potentially useful idea of storing relevant product information in a vector though.

The visual images of the **spatial signature analysis** - technique (SSA) can not be captured by binary sensors either. Although continuous sensors can be installed on a manufacturing line for this purpose, most processes that are currently monitored in a binary way do not possess the necessary characteristics to make SSA possible. The analysis in SSA is based on phenomena that are clearly visible and distinguishable in the 2-dimensional plane. This type of phenomena can only be found in processes of the 2D oriented *surface processing operations* nature (see Figure 2.3 for reference). For processes that possess a different nature, such as the incandescent lamps production lines, it has proven to be far too difficult to *automatically* extract meaningful features from product images. It is possible however to use this approach in a targeted way as in common Vision appliances.

The continuous waveform signals that are essential for the **diagnostic feature extraction** approach can not be captured by binary sensors. If, for the process under consideration, such an indicative process parameter is available, it is advisable to investigate the possibilities of installing continuous sensors to monitor it. For many processes, among which the incandescent lamps production lines, such a dominating process parameter does not exist however. Therefore, the **semi-direct classification** techniques can not be used for binary sensors either.

The final analysis technique, **Statistical Process Control** (Section 3.2.4), also seems to rely on the availability of continuous data. Normally, direct *product* or *process* parameters that indicate the quality level of the process are compared to a standard. This direct continuous data is not available for binary sensors however. As discussed in Section 4.1, each datapoint represents the presence of a certain characteristic A. Whatever this characteristic precisely represents, and whatever the consequences of the (lack of) possession of it, this means that *events* are counted. Three cases are possible:

In the case of a process parameter the characteristic is directly informative on the state of the *process*, and therefore is readily available as a *feature* that can be diagnosed as a next step.

Product characteristics are informative on the product itself however. From a process quality control - point of view the main concern is that the *process* failed to provide one of the products with the needed characteristic. If this is extremely rare and immediately alarming, this is clearly an interesting *feature*.

However, if it is *normal* for the production process under consideration that some products fail to possess a certain characteristic, and therefore maybe need to be rejected, one blemish in itself is *no feature* yet. Clearly, an analysis is needed of the amount of such blemishes in a certain period. If an amount of blemishes can be found that corresponds to a situation in which root-causes are present, exceeding this level for a certain sensor in a certain period would form an informative feature of the system. In the next Section the concept of *entitlement* will be introduced to calculate these kind of levels.

If this entitlement calculation can be developed, the occurrence of features for all three cases is well defined. Then, these adapted SPC techniques will form a very useful analysis tool to transform binary data into informative features. This makes indirect classification based on these features possible. Therefore, this analysis tool is chosen. The calculation of the entitlement level and further adaptations for this analysis technique will be developed in Section 4.3.

4.2.2 Classification techniques

Experiments for the practical case of incandescent lamps production lines have indicated that the relationships between sensor values and root-causes in this binary case are too complex for direct classification. Furthermore, efforts to create linear and higher order decision boundaries as illustrated in Section 3.3.3 have failed. The noise ratio seems to be too high to allow any direct pattern recognition approaches. This underlines the usefulness of the entitlement calculation approach. The analysis step has an important data pretreatment role to fulfill. The classification step can then be based on the features delivered by the adapted SPC analysis tool. These features have a binary nature, since they are either present or absent. Several of the discussed indirect classification techniques are able to classify cases with these features as input.

Decision boundaries are not very suitable for this type of data however. Furthermore, they possess the disadvantage that features can not easily be weighted in a different manner. Fuzzy techniques are possible, but can be unnecessarily complicated.

Neural networks can establish a relationship based on training cases which is black box like. The objective of the classification tool is to deliver root-causes to the operators, technicians and process engineers for manual feedback. It is therefore especially important to establish an insightful classification structure instead of a black box.

Learning algorithms therefore seem to be the best option for the current development of a binary process quality control methodology. Their insightful structure helps in identifying the established link between training cases, their features and the occurring root-causes. Furthermore, comparison of a new case to its most similar training case yields a very transparent overview of similar and different features for both cases. This provides the opportunity to investigate misclassification with engineering knowledge.

Training efforts will be very dependent on the frequency of occurrence of features and the amount of distinct root-cause classes that the tool needs to be trained on. It will be situation dependent which classification technique requires most training to be accurate. For the Philips case the training efforts involved in employing the learning algorithm are expected to be manageable, therefore this technique is chosen for the current development.

4.3 Analysis technique development

4.3.1 Entitlement calculation

As the objective of process quality control is maintaining the state of control in which only *normal* causes are present, a blemishes-level needs to be found that differentiates between situations in which only *normal* causes are present and those in which *assignable* causes occur. As such, a standard is developed to which a new datapoint can be compared to assess whether a problem with a certain root-cause is present for the new point. To refer to this level the term *entitlement* level is introduced.

entitlement level - the average blemishes level for production periods in which only normal causes are present

This level therefore indicates the best possible quality level for the process given its current design and its current state. The term *entitlement* is used to express that this level allows to entitle the state of the process as normal or abnormal. The calculation of the entitlement level will yield a distribution that describes the chance of encountering every possible count level[2], if only *normal* causes are present. The upper limit of this distribution therefore indicates a count level beyond which, with a high level of significance, an assignable cause is present. Comparing values to this upper limit is termed an *entitlement check.*

entitlement check - determining whether a datapoint possesses assignable causes with a high level of significance. This is performed by comparing the datapoint to the upper limit that is linked to the entitlement level and classifying it as abnormal if it lies above the upper limit

Initially, a dataset is needed for the calculation of the entitlement level. Afterwards, new datapoints can be submitted to the entitlement check, based on the value for the upper limit that was concluded from the entitlement calculation to represent the normal datapoints distribution.

Wheeler [Whe95] discusses a very interesting common property of all *homogeneous* datasets which can be used as the core of the calculation of the entitlement level. In this respect homogeneity means that all datapoints are random samples of the same, stable, underlying probability distribution function [Whe95].

property 4.1 - if a dataset is *homogeneous*, more than 99% of all datapoints lie within the $\widehat{\mu} - 3 \cdot \widehat{\sigma}$ and the $\widehat{\mu} + 3 \cdot \widehat{\sigma}$ limits, no matter of what type the underlying probability distribution is

Note that the definitions of $\widehat{\mu}$ and $\widehat{\sigma}$ are dependent upon the type of data and the knowledge of the underlying probability distribution function, as will be discussed at the end of this Session.

Property 4.1 is illustrated for a variety of distributions in Figure 4.1.

[2]In some cases this count level needs to be corrected for the corresponding area of opportunity, see Section 4.3.3. The illustrated entitlement calculation procedure, although then based on these corrected *count levels per unit of area of opportunity*, remains unchanged however.

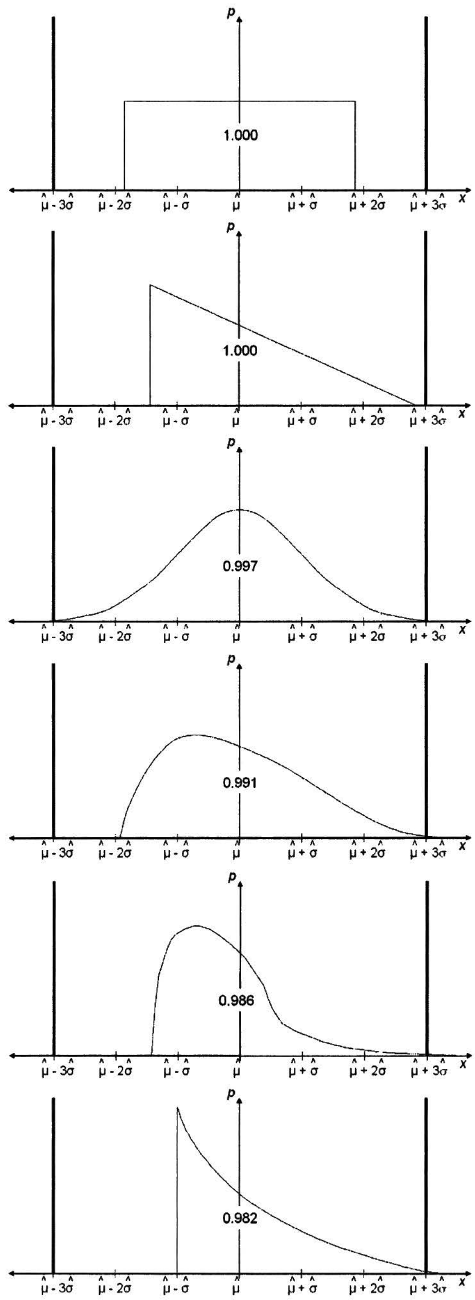

Figure 4.1: Different homogeneous probability distributions (adapted from [Whe95]) showing property 4.1. The values inside the distribution functions indicate the exact fraction of points inside the limits for these examples

The state of the production process in which only normal causes are present does not vary over time and therefore forms a stable underlying generator for homogeneous datapoints. This normal causes distribution, indicated by "N" in Figure 4.2, generates the datapoints for situations in which only normal causes are present in the production line. Therefore, typically relatively low count levels result from this distribution. In practice, the process can get out of control though, leading to an abnormal situation in which assignable causes are present. As a result, a different underlying distribution generates the higher count levels that are expected for these datapoints with assignable causes (indicated by "A" in Figure 4.2).

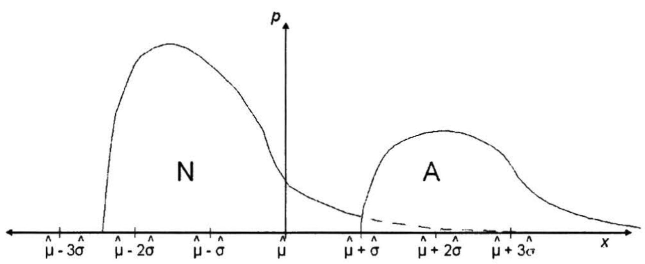

Figure 4.2: The process distribution in practice actually exists of a separate distribution of counts when only normal causes are present (N) and one for the assignable causes (A)

If property 4.1 is applied in a repetitive manner, the datapoints in the initial dataset can be identified as normal or abnormal. Hereto, the entitlement level ($\widehat{\mu}$) and upper limit ($\widehat{\mu} + 3 \cdot \widehat{\sigma}$) of the initial practical dataset need to be calculated. Figure 4.3 shows the result.

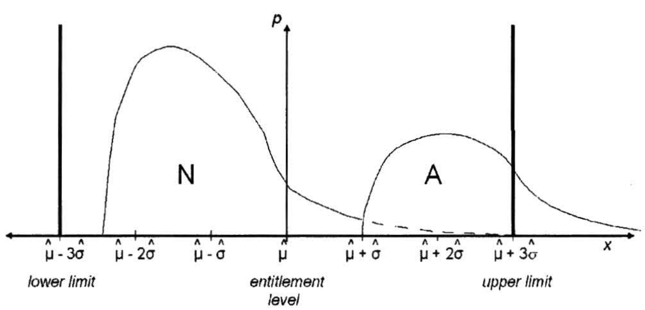

Figure 4.3: The practical process distribution with first estimates of entitlement level and limits. Those points that lie above the upper limit are identified as belonging to the abnormal causes - distribution

As the abnormal datapoints and normal datapoints come from a different probability distribution function the initial practical dataset is not homogeneous, and therefore more than 1% of the datapoints is expected to lie above the upper limit. Indeed, the practical case of Philips Lighting shows that in practice much higher percentages are found. The datapoints with high count levels are very likely to have been generated by the abnormal causes distribution. Therefore, the datapoints that appear to lie above the upper limit are identified as abnormal. Filtering these datapoints from the dataset (illustrated by colouring them gray) leaves a practical dataset with a higher fraction of normal datapoints than the initial set. As a consequence, the entitlement level and upper limit of this filtered dataset will be lowered, as illustrated in Figure 4.4.

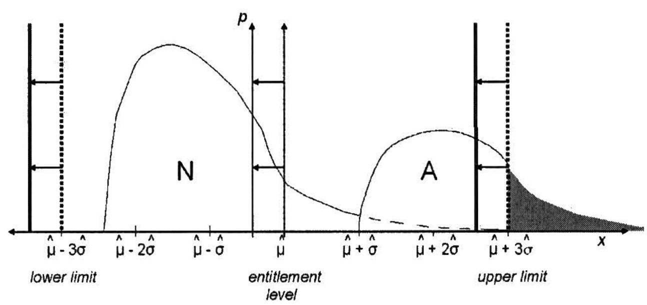

Figure 4.4: Filtering out some abnormal datapoints lowers the entitlement level and upper limit

The result is shown in Figure 4.5. Again, if more than 1% of the datapoints lies above the $\widehat{\mu} + 3 \cdot \widehat{\sigma}$ limit, this is interpreted as a signal that the dataset is not homogeneous yet. Therefore, the abnormal points that lie above this upper limit are filtered out once more.

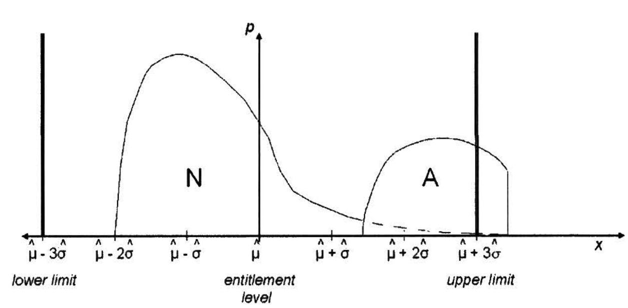

Figure 4.5: Still more than 1% of the datapoints lies above the upper limit. The dataset is therefore still not homogeneous

This process needs to be repeated several times (see Figure 4.6) until only less than 1% of the datapoints lies above the $\widehat{\mu} + 3 \cdot \widehat{\sigma}$ limit. Then, a homogeneous dataset with only the normal datapoints remains. This final result is illustrated in Figure 4.7.

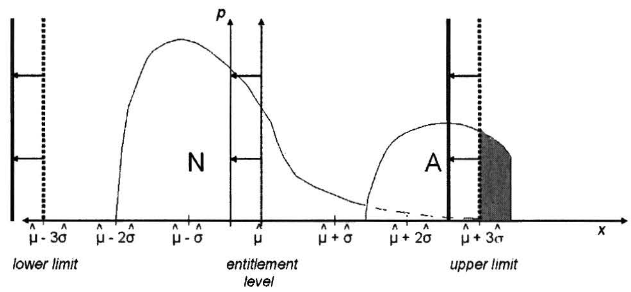

Figure 4.6: Another filtering cycle in which abnormal points are filtered out

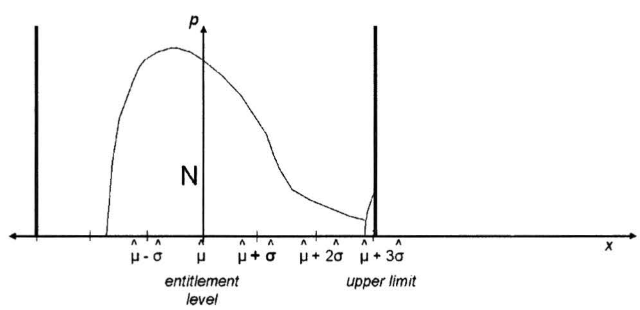

Figure 4.7: Final result: A homogeneous dataset with only the normal datapoints without assignable causes. The real entitlement level and value for upper limit have now been found

It is important to realize that the distinction between normal and assignable causes is in some way virtual. As discussed in Section 3.2.4, assignable causes are external to the process design. No process is designed robust enough to eliminate all possible problems however. The focus is not on pinpointing the exact moment that a relatively small cause that is always present to some extent, becomes an assignable cause that can be taken away. In stead, the distinction serves as a practical trigger to come into action.

By plotting the histogram of the initial practical dataset from the binary sensors it can be assessed whether a clear distinction between normal and abnormal cases as illustrated in Figure 4.2 indeed appears to exist. Furthermore, monitoring the speed of convergence, measured as the amount of filtering cycles (and amount of filtered points per cycle) needed to reach a homogeneous dataset, can help in assessing whether or not the distinction is clear for the practical case under consideration. Hereto, the production periods that are characterized as abnormal should be diagnosed to assess whether, in most cases, relevant assignable causes are indeed present. Furthermore, it is important to check whether identifiable root-causes are indeed absent for periods that were characterized as normal.

The entitlement calculation needs to be executed for every binary sensor signal. To reduce future numerical efforts an automated entitlement calculation tool has been developed. This tool retrieves data from the control system of a manufacturing line via ActiveFactory, places it in an Excel spreadsheet and performs the entitlement calculation procedure. It ultimately yields the entitlement of each datapoint as normal or abnormal, presents an overview of the filtering behavior and indicates the descriptive statistics of the practical distribution.

As stated, the specific way of calculating the estimators $\widehat{\mu}$ and $\widehat{\sigma}$ that define the entitlement level and upper limit in this entitlement calculation depends upon the specific type of data. Two important categories can be distinguished:

Data in subgroups

If the binary sensor data in a specific situation can be rationally categorized in subgroups, such as batches of products, the data is characterized as subgroup data.

The definitions of $\widehat{\mu}$ and $\widehat{\sigma}$ for subgroup data are given by these formulas [Whe95]:

$$\widehat{\mu} = \overline{\overline{x}} = \frac{\sum_{i=1}^{n_s} \overline{x_i}}{n_s} \tag{4.1}$$

where:

$\overline{x_i}$ = subgroup average for the i^{th} subgroup

n_s = number of subgroups

The size and frequency of these subgroups should be chosen such that the likeliness that process changes will occur between the samples, rather than within the samples, is maximized [Whe95]. This procedure is relatively subjective though.

Furthermore:

$$\widehat{\sigma} = \frac{\overline{R}}{d_2 * \sqrt{s}} \tag{4.2}$$

where:

$s =$ the size of a subgroup

$d_2* =$ the bias correction factor for using average ranges to estimate variances. Its value is dependent upon the size of the subgroup s and the amount of subgroups n_s [Whe95].

and

$\overline{R} =$ average range of the subgroups, according to:

$$\overline{R} = \frac{\sum_{i=1}^{n_s} R_i}{n_s} \tag{4.3}$$

where:

$R_i =$ range of the i^{th} subgroup

$n_s =$ number of subgroups

Data in time series

If the data from the binary sensors is ordered in time, and no rational subgroups can be formed, the data is characterized as time series of data. The best method to estimate $\widehat{\mu}$ and $\widehat{\sigma}$ for this type of data is suggested by Wheeler [Whe95] to be:

$$\widehat{\mu} = \overline{x} = \frac{\sum_{i=1}^{n_d} x_i}{n_d} \tag{4.4}$$

where:

x_i = value for i^{th} datapoint

n_d = number of datapoints

Furthermore:

$$\widehat{\sigma} = \frac{\sum_{i \gg j=1}^{n_d-1} R_{i \gg j}}{n_d - 1} \tag{4.5}$$

where:

$R_{i \gg j}$ = the two-point moving range between datapoint i and datapoint j

n_d = number of datapoints

Depending on the type of data in the initial practical dataset formulas 4.1 and 4.2 or 4.4 and 4.5 can be used to estimate $\widehat{\mu}$ and $\widehat{\sigma}$, for the execution of the entitlement calculation.

4.3.2 Fitting a distribution

Upon completion of the entitlement calculation the normal datapoints have been distinguished. If a theoretical distribution function exists that matches the practical normal causes - distribution, the μ and σ of this distribution can be used to perform the *entitlement check* for future datapoints in stead of using their estimates $\widehat{\mu}$ and $\widehat{\sigma}$. This entitlement check for new datapoints is described in Section 4.3.4.

Using the values μ and σ of the distribution is beneficial because the best generalization to new data is obtained when the mapping captures the underlying systematic aspects of the data rather than the irregularities present in the practical dataset (which are therefore present in the estimators $\widehat{\mu}$ and $\widehat{\sigma}$ as well) [Bis03].

Furthermore, the entitlement check can then be based on those statistics that are representative for the specific type of probability distribution. As illustrated in Section 4.3.3 especially $\widehat{\sigma}$ is very dependent upon the specific distribution.

Literature offers some very clear methods that guide in finding the theoretical distribution that fits the practical dataset of one of the sensors (e.g. [Law01, Whe95]).

A thorough approach can consist out of the following steps (for a more elaborate description the reader is referred to [Law01]):

Assess sample independence

Various statistical techniques, used in finding a fitting theoretical distribution, require that the data $x_1, x_2, ..., x_n$ are an independent sample from some underlying distribution. Correlation plots and scatter diagrams are easy and insightful tools to assess the validity of this assumption.

Hypothesize families of distributions

Prior knowledge on the type of data can be used to assess possible distributions on theoretical grounds (e.g. untreated binary data has a discrete character). Furthermore, summary statistics of the dataset can be insightful. Examples are the range of the data, the mean, variance and skewness. Also, alternative measures for variability like the coefficient of variation cv for continuous data and the lexis ratio for discrete data can be insightful. Next, histograms can form an insightful graphical estimate of the plot of the theoretical probability distribution. Finally, quantile summaries and boxplots can be used.

Estimation of parameters

Specifying a distribution finally involves the estimation of the specific parameters of the distribution. Maximum-likelihood estimators (MLE) are a suitable tool to specify an estimator for a particular parameter.

Determining quality of fit

In order to determine how representative the fitted distributions are several methods exist. First, heuristic tools like density/histogram overplots, frequency comparisons, distribution function difference plots and probability plots can be used. Next, goodness-of-fit tests like the chi-square test and the Kolmogorov-Smirnov test are insightful to assess the degree of conformance between the practical data and the theoretical distribution.

4.3.3 Binomial and Poisson distribution

Two often encountered theoretical distributions for binary sensors are Binomial and Poisson because of the discrete character of the counts. Whether the applicable theoretical distribution in a specific situation is Binomial or Poisson is dependent on the type of area of opportunity, which is defined as:

area of opportunity - the maximum possible amount of blemishes that serves as a reference to which the amount of occurring blemishes can be related

For some characteristics, several blemishes can take place inside one semi-product. The area of opportunity is then formed by the maximum amount of blemishes per product. Alternatively, if a semi-product can have one blemish at most, it is interesting how many of the semi-products possess such a blemish.

In the former case the area of opportunity is as abstract as some finite region of space, time, or product [Whe95], whereas it is as concrete as the number of discrete items in the latter. For the former case Binomial distributions may apply whereas Poisson may be applicable to the latter case.

The area of opportunity can either be the same for all datapoints, or differ from period to period as is usually the case with, for example, the amount of produced products. For fixed area of opportunity the entitlement level can be based on the count level (x_{ENT_i}). For situations in which the area of opportunity is different for each datapoint, the entitlement level needs to be adjusted to counts per unit of area of opportunity (r_{ENT_i}) to allow meaningful comparison of the datapoints. The same distinction exists for the upper limit.

For the **binomial distribution**, the following conditions need to apply:

- the area of opportunity for the count must consist of n_i distinct items
- each of the n_i distinct items must be classified as possessing, or not possessing some characteristic A

- the probability p_A that an item has the counted attribute must be the same for all n_i items in one sample. The value of p_A should only differ *between* samples, not *within* samples
- the events (items not possessing characteristic A) are independent of each other

If these conditions are satisfied, and a binomial distribution appears to provide a good fit to the normal causes - dataset, the following formulas apply:

For fixed area of opportunity

$$x_{\mathrm{ENT}_i} = \mu = n_i \cdot p_A \tag{4.6}$$

where:

x_{ENT_i} = entitlement level for counts, for datapoint i

p_A = the probability that an item possesses attribute A

n_i = the amount of examined items

$$x_{\mathrm{UL}_i} = \mu + 3 \cdot \sigma = n_i \cdot p_A + 3 \cdot \sqrt{n_i \cdot p_A \cdot (1 - p_A)} \tag{4.7}$$

where:

x_{UL_i} = upper limit for counts, for datapoint i

For varying area of opportunity

$$r_{\mathrm{ENT}_i} = \mu = p_A \tag{4.8}$$

where:

r_{ENT_i} = entitlement level for counts per unit of area of opportunity, for datapoint i

$$r_{\mathrm{UL}_i} = \mu + 3 \cdot \sigma = p_A + 3 \cdot \sqrt{\frac{p_A \cdot (1 - p_A)}{n_{i_t}}} \tag{4.9}$$

where:

r_{UL_i} = upper limit for counts per unit of area of opportunity, for datapoint i

n_{i_t} = the amount of examined items for the sample of time period t

For the **poisson distribution**, the following conditions need to apply [Whe95]:

- the counts are counts of discrete events. These counts are the blemished products x_{i_t}, where i indicates the sensor and t the time period
- the discrete events occur within a finite region. This finite region is the area of opportunity for the count and is formed by the amount of semi-products p_{m_t} that are being produced on machine m in time period t
- the events occur independently of each other (memoryless property [Ros00]), and the likelihood of an event is proportional to the size of area of opportunity (i.e. the more semi-products are being produced, the more blemished products x_{i_t} are likely to occur)
- the events are rare (e.g. $x_{i_t} < \frac{1}{50} \cdot p_{m_t}$ (i.e. less than 2% of produced semi-products are blemished))

If the stated conditions apply and the Poisson distribution appears to provide a good fit to the practical dataset, the following formulas can be used to calculate the entitlement level and the upper limit:

For fixed area of opportunity

$$x_{\mathrm{ENT}_i} = \mu = \frac{\sum_{t=1}^{n_t} x_{i_t}}{n_t} \tag{4.10}$$

where:

x_{i_t} = the count of rejected products by sensor i during period t

n_t = the amount of periods

$$x_{\mathrm{UL}_i} = \mu + 3 \cdot \sigma = \frac{\sum_{t=1}^{n_t} x_{i_t}}{n_t} + 3 \cdot \sqrt{\frac{\sum_{t=1}^{n_t} x_{i_t}}{n_t}} \tag{4.11}$$

For varying area of opportunity

If the area of opportunity varies from time period to time period, as is normally the case with the amount of products produced per period of fixed length, then the formulas become:

$$r_{\mathrm{ENT}_i} = \mu = \frac{\sum_{t=1}^{n_t} x_{i_t}}{\sum_{t=1}^{n_t} p_{m_t}} \tag{4.12}$$

$$r_{\mathrm{UL}_i} = \mu + 3 \cdot \sigma \quad = \quad \frac{\sum_{t=1}^{n_t} x_{i_t}}{\sum_{t=1}^{n_t} p_{m_t}} + 3 \cdot \sqrt{\frac{\frac{\sum_{t=1}^{n_t} x_{i_t}}{\sum_{t=1}^{n_t} p_{m_t}}}{p_{m_{t_{new}}}}} \tag{4.13}$$

where:

$p_{m_{t_{new}}}$ = the amount of products produced on machine m in the new period t_{new}

In equation 4.13 for the upper limit, the denominator contains the term $p_{m_{t_{new}}}$ which is dependent upon the new period t_{new}. The upper limit is therefore different for each datapoint, depending on the amount of products produced.

4.3.4 Entitlement check

The entitlement check aims at deciding whether a certain sensor value, of a new production period, is normal or abnormal. Hereto, it uses the knowledge of the normal causes - distribution.

The transformation of the value of sensor i for time period t, x_{i_t}, into a binary feature bf_{i_t} relating to the same sensor and period is executed as follows:

$$\begin{aligned} &\text{If } x_{i_t} > x_{\mathrm{UL}_i}, \text{ then } bf_{i_t} = 1 \\ &\text{If } x_{i_t} \leq x_{\mathrm{UL}_i}, \text{ then } bf_{i_t} = 0 \end{aligned} \tag{4.14}$$

where:

x_{i_t} = amount of counts for sensor i, in time period t

x_{UL_i} = upper limit of counts of normal causes - distribution, for sensor i

As discussed, in situations of varying area of opportunity per datapoint, the amount of counts needs to be related to the corresponding area of opportunity, the transformation then becomes:

$$\begin{aligned} &\text{If } r_{i_t} > r_{\mathrm{UL}_i}, \text{ then } bf_{i_t} = 1 \\ &\text{If } r_{i_t} \leq r_{\mathrm{UL}_i}, \text{ then } bf_{i_t} = 0 \end{aligned} \tag{4.15}$$

where:

r_{i_t} = amount of counts per unit of area of opportunity for sensor i, in time period t

r_{UL_i} = upper limit of counts per unit of area of opportunity of normal causes - distribution, for sensor i

The value for x_{UL_i} or r_{UL_i} is preferably calculated with the use of a well fitting theoretical distribution:

$$x_{\mathrm{UL}_i} = \mu + 3 \cdot \sigma \tag{4.16}$$

or

$$r_{\mathrm{UL}_i} = \mu + 3 \cdot \sigma \tag{4.17}$$

With μ and σ the parameters of the found theoretical distribution.

Note that in the case of a Poisson distribution the upper limit r_{UL_i} is different for each time period. See Section 4.3.3 for the discussion on this time dependency of the upper limit $r_{\mathrm{UL}_{i_t}}$.

Alternatively, if no fitting distribution could be found, $\widehat{\mu}$ and $\widehat{\sigma}$ of the last filtering cycle of the entitlement calculation procedure need to be used as estimators of the parameters μ and σ in *estimating* x_{UL_i} or r_{UL_i} for the entitlement check:

$$\widehat{x}_{\mathrm{UL}_i} = \widehat{\mu} + 3 \cdot \widehat{\sigma} \tag{4.18}$$

or

$$\widehat{r}_{\mathrm{UL}_i} = \widehat{\mu} + 3 \cdot \widehat{\sigma} \tag{4.19}$$

As such the developed methodology is able to indicate, real-time, whether a phenomenon has occurred for each time period. If binary features (bf_{i_t}) are present, the analysis methodology is also able to pinpoint the location (indicated as i) and time (t) of occurrence. This data is printed on screen and the system state is therefore ready to be classified in the next quality control step.

4.4 Classification technique development

As discussed in Section 3.3.3 learning algorithms can compare a new case to diagnosed training cases. Hereto, a certain set of features is predefined. For each training case, the algorithm then compares each of the features of the new case to the training case its features. As discussed, the developed analysis tool delivers binary features. The first step (Section 4.4.1) is therefore to select the most relevant binary features as input for the comparison algorithm.

An overview of several learning algorithms that can handle binary features is presented by [Win98]. None of the discussed algorithms is readily suitable for the quality control purpose, however. Therefore, in Section 4.4.2 a new algorithm will be developed, based on discussed algorithms by Winiwarter [Win98].

4.4.1 Binary feature selection

The aim of the comparison algorithm is to compare system states. A learning algorithm is able to compare vectors that contain features of the system, if presented in a binary way. These system state vectors $\mathbf{s}_t$ for each time period t therefore need to possess the following kind of structure:

$$\boldsymbol{s}_t = \begin{bmatrix} bf_{1_t} \\ bf_{2_t} \\ bf_{3_t} \\ \cdot \\ \cdot \\ \cdot \\ bf_{n_t} \end{bmatrix} = \text{e.g.} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$$

Figure 4.8: Structure of system state vector

The phenomena that are delivered by the analysis technique are very informative on the system state. The first set of binary features of the system state vector is therefore formed by the results of the entitlement check for every single sensor, for the past production period. Hereto, the bf_{i_t}'s that are delivered by the analysis tool can directly be used.

Are there any other features of the system state known, that can improve the system state description? It was illustrated in the state space modeling approach that engineering knowledge can be helpful in describing the system state. Given the engineering knowledge of the specific production process under consideration, some mutually occurring abnormal values may typically point to another root-cause than in the case of a single abnormal value. As comparison algorithms assess the similarity between cases based on the *amount* of similar binary features (in a weighted manner), the inclusion of binary features that represent the occurrence of such correlations can improve the classification correctness rate. Therefore, a second interesting set of binary features of the system state vector represents whether or not these typical correlations are occurring for the case under consideration.

Assuming that such a special correlation exists between the sensor values of sensor i and sensor j, this second set of binary features is determined as follows:

$$\begin{aligned} &\text{If } bf_{i_t} = 1 \text{ and } bf_{j_t} = 1, \text{ then } bf_{q_t} = 1 \\ &\text{If } bf_{i_t} = 0 \text{ or } bf_{j_t} = 0, \text{ then } bf_{q_t} = 0 \end{aligned} \tag{4.20}$$

where:

bf_{i_t} = occurrence of a phenomenon for sensor i

bf_{q_t} = q^{th} binary feature of the system state vector

Analogously, it is possible that engineering knowledge suggests that an abnormal sensor value x_k, while sensor value x_l is normal, points towards a specific root-cause. The corresponding binary feature bf_r is then determined as:

$$\begin{aligned}&\text{If } bf_{k_t} = 1 \text{ and } bf_{l_t} = 0, \text{ then } bf_{r_t} = 1\\&\text{If } bf_{k_t} = 0 \text{ or } bf_{l_t} = 1, \text{ then } bf_{r_t} = 0\end{aligned} \tag{4.21}$$

A third and final category of binary features that can give information on the system state is formed by occurring events. Examples of possibly interesting events are:

- stops
- breakdowns
- change overs
- shift identification tags
- maintenance activities

If such an event, v, occurs the corresponding $bf_{v_t} = 1$, if not, $bf_{v_t} = 0$. The descriptive power of these events lies in the fact that their occurrence can increase the likelihood of the presence of a certain root-cause.

Some problems, like bad adjusted machine parts, typically occur more often after maintenance activities or change overs, during which they are often replaced.

Because of the different way of working of different shift teams, some problems may occur more often for a certain team. This explains the descriptive power of shift identification tags regarding the prediction of the presence of certain root-causes.

4.4.2 Learning algorithms

The core of the algorithm is to compare each of the discussed binary features of the system state vector $\mathbf{s}$ of the new case, denoted by $\mathbf{s}_{new}$, with the features of the training state vector $\mathbf{s}_{tr}$. Based on whether both cases contain bf_i or not, a certain measure of similarity needs to be constructed. This allows the selection of the most similar training case after iterating through all cases $\mathbf{s}_{tr_t}$.

$$
\begin{aligned}
SIM(\mathbf{s}_{new}, \mathbf{s}_{tr}) &= \sum_{i=1}^{n_f} \sigma(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) \\
&- \sum_{i=1}^{n_f} \delta_{new}(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) \\
&- \sum_{i=1}^{n_f} \delta_{tr}(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i})
\end{aligned} \tag{4.22}
$$

where:

n_f = amount of features

- $\sigma(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) =$

  1 if $\mathbf{s}_{new_i} = 1 \wedge \mathbf{s}_{tr_i} = 1$

  0 otherwise

- $\delta_{new}(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) =$

  1 if $\mathbf{s}_{new_i} = 1 \wedge \mathbf{s}_{tr_i} = 0$

  0 otherwise

- $\delta_{tr}(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) =$

  1 if $\mathbf{s}_{new_i} = 0 \wedge \mathbf{s}_{tr_i} = 1$

  0 otherwise

Winiwarter [Win98] discusses several methods of weighing the binary features. For the current classification purpose none of the discussed weighing methods is completely suitable however.

The core of the classification principle is that the system state will typically contain certain binary features for certain root-causes. It is therefore possible to assign an importance factor to each feature for each root-cause class. A simple but effective parameter is formed by the proportion of training cases that possess bf_i in class j. Classification correctness can then be improved by weighing the correspondence and difference between present features according to this feature - importance parameter.

With this extension, equation 4.22 becomes:

$$
\begin{aligned}
SIM(\mathbf{s}_{new}, \mathbf{s}_{tr}) &= \sum_{i=1}^{n_f} \sigma(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) \cdot \pi_{ji} \\
&- \sum_{i=1}^{n_f} \delta_{new}(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) \cdot (1 - \pi_{ji}) \\
&- \sum_{i=1}^{n_f} \delta_{tr}(\mathbf{s}_{new_i}, \mathbf{s}_{tr_i}) \cdot \pi_{ji}
\end{aligned}
\tag{4.23}
$$

where:

π_{ji} = the proportion of cases in class j that contains feature i

j = the class of the training case $\mathbf{s}_{tr}$

4.4.3 Training the tool

The comparison algorithm can only be used after some thorough training. The system state vectors of training cases are automatically delivered by the developed analysis tool. Training the diagnosis tool therefore consists of linking the present root-causes to these system state vectors.

The presence of root-causes in the training cases can be investigated by several other tools. The principle of diagnosing pictures of semi-products that was part of the spatial signature analysis - technique seems very interesting to diagnose problems. Depending on the specific production process, the pictures taken at those moments and places at which a problem is occurring may be very useful to find the root-causes and therefore train the classification tool. Another possibility suggested by spatial signature analysis was the use of operator knowledge. A final possibility is the use of documentation. Problems that have been fixed by maintenance personnel can be investigated to yield their root-causes.

4.4.4 Testing and use

The final step of the development of the binary process quality control methodology consists of testing. The classification correctness rate needs to be assessed by comparing the prediction of present root-causes by the tool to their actual presence.

Performance optimization can be achieved by altering the content of the system state vector. Certain informative binary features can be added, while less informative features can be excluded. Extra training of the tool might help in improving the classification correctness.

Depending on the specific situation, the emphasis will be on the reduction of false alarms or on the achieved sensitivity of the tool.

After the performance of the tool has been optimized the automated methodology can be put into use by coupling the relevant parts of it to the information stream that comes from the binary sensors. An exemplary procedure of all these steps is described in Chapter 5.

4.5 Summary

The aim of this Chapter was the development of a binary process quality control methodology. After discussing the specific properties of binary sensor data the consequence of using this type of data for the applicability of the continuous techniques of Chapter 3 was discussed. SPC techniques proved useful in determining features for some cases (process errors and *rare* product errors) but a decision procedure needed to be developed for *common* product errors. For this purpose the entitlement calculation was developed, which is able to identify each of the training datapoints as normal or abnormal. Furthermore, the latter points are automatically filtered out.

Next, a procedure was described for fitting a distribution that represents the resulting set of normal datapoints, and the specific properties of Binomial and Poisson distributions (often encountered in the case of discrete count data) were discussed. Section 4.3.4 discussed the subsequent determination of abnormality for new datapoints, which is preferably based on the parameters of the found distribution.

Then, a classification tool for these new datapoints was developed. Indirect classification appeared most suitable, and therefore classification was based on comparison to previously diagnosed training cases. Basically the amount of similar binary features for the system state vector of the new case and of each of the training cases is assessed in a weighted manner.

The developed methodology is therefore based on the characterization of new datapoints by evaluation of the possession of several binary features, a subsequent comparison to diagnosed training cases, and finally a link to the root-cause of its most similar training instance.

Chapter 5

Customizing and testing the developed methodology

The final step, after the development of the binary process quality control methodology, is the customization to a specific production process. As mentioned earlier, the production process of incandescent lamps at Philips Lighting will serve as the practical platform.

In Section 5.1.1 an overview of this production process is presented. As the automated process quality control techniques that are currently in use are limited, the current application is expected to yield good results.

As a first step, attention is devoted to the monitoring of the process, which is performed by binary sensors.

Next, the application of the developed analysis techniques to this data are discussed. After the calculation of the entitlement levels, it is tried to characterize the normal datapoints by a theoretical distribution. For the training period of one month the entitlement check then delivers the occurring abnormal values.

The classification step, as developed in Section 4.4, is then customized to the incandescent lamps process. Firstly, the content of the state vector is deducted by selecting relevant binary features. Next, for the training period, the present root-causes are determined by employing a camera tool, especially developed for this purpose. The final customization step, the actual training of the classification tool, consists of linking these root-causes to their state vectors.

In Section 5.6, the customized tools are then put into use for the first time. The predictions of present root-causes as yielded by the combined efforts of the analysis and classification tools are then compared to the actual root-causes. This information is again retrieved by employing the camera tool. The correctness of the classification is then finally assessed.

5.1 Philips Lighting

5.1.1 Production process of incandescent lamps

Philips Lighting is a leader in the lighting industry. Yearly about 2 billion incandescent and 0.8 billion fluorescent lamps are produced by Philips in many factories in different parts of the world. The production process of these lamps can predominantly be characterized as "assembly" of semi-products. At many points in a production line there are sensors that check for compliance with a certain requirement and produce a binary output accordingly. As such, semi-products are approved or rejected by these sensors. If a product is rejected it means that components are lost and because 70 %[1] of the cost price of a lamp is formed by its components, rejects must be avoided as much as possible, especially near the end of the line.

Lamp production lines consist of a number of indexing turrets that are coupled via control transports and buffer transports. The production system can be characterized as automated, indexed, processing high volumes and quite robust.

The product types that are under consideration in this research project are those that are most common and have the largest production volumes, the incandescent lamps.

Production of incandescent lamps for Philips is done on more than 200 production lines which are very much similar. Production lines consist of coupled machines that are indexing turrets with rigid mechanization driven by cams. The turrets are mills with a certain amount of product holders (depending on the machine) on the outside. In this way access for all operations is ensured, and because the mills are indexing, products can pass various production stages on one machine.

Most (standard) lamps are produced on Philips B-groups that have a production speed of 4500 prod/hr. On a B-group there are approximately 25 sensors to identify rejects. Siemens S5 or S7 is used as control system (PLC) on most of the B-groups for synchronizing the machines and for rejecting defect semi-products. Some of the lines have been extended with a monitoring system that stores the sensor signals from the control system.

5.1.2 Characterization of production process

Following the categorization of Section 2.2 the production process of Philips Lighting needs to be assessed in terms of its *nature*, *structure*, and its *costs* of faulty products.

The *nature* of the process is a combination of assembly operations and some processing operations (see Figure 2.3). Most assembly operations belong to the subcategory of permanent joining processes (e.g. melting the stem into the bulb, and threading), with some exceptions belonging to the mechanical fastening type (e.g. placement of a coil).

[1] this is the percentage for incandescent lamps

The processing operations consist of some shaping processes (e.g. bending the lead-in-wires) and some property enhancing operations (e.g. heat treatments for the bulb).

The *structure* of the production process is the connected line flow structure (type III in Figure 2.4). The *costs* associated with producing nonconforming products are relatively low (compared to the most advanced industries).

This type of production process that is mainly categorized by an assembly nature performed in a continuous flow structure is often observed and is therefore a useful example.

5.1.3 Current process quality control techniques

Current practice at Philips is that only the monitoring process is automated. Sensors check the compliance of the product with a certain condition and produce a binary output accordingly. The subsequent analysis and diagnosis phase, as well as the feedback, are based on the experience of the operators, technicians and process engineers, rather than following automated methodologies that make use of modeling. It is therefore expected that starting to make use of the data that is produced by the monitoring system can greatly reduce the current amount of rejected products.

5.2 Monitoring the process

In figure 5.1 an overview of the B-group configuration is given. The different machines are indicated as well as the positioning of the sensors that check product compliance.

5.2.1 Stem making machine

In the stem making machine (indicated as SMM in the figure) the first components are melted together to make a stem. A flare, which is a small hat-shaped glass product with a hole in it is put in the product holder. Next two lead-in-wires are inserted into this open flare. Finally an exhaust-tube which has the form of a long (11 cm) hollow glass tube is pushed through the hole in the flare.

The first sensor in the line checks the presence of the exhaust-tube by a simple mechanical action. If this tube appears to be absent the product will not be transferred at the last position of the mill but will in stead be thrown into a reject-bin. The flare and exhaust-tube are now heated above the glass melting temperature T_m. Two squeezing blocks clamp the flare from two sides, as such melting it together with the exhaust-tube that is going through it. The result is a flattened piece in the middle of the semi-product in which two holes are blown in order to open the access to the exhaust-tube channel for a pumping process later in the line.

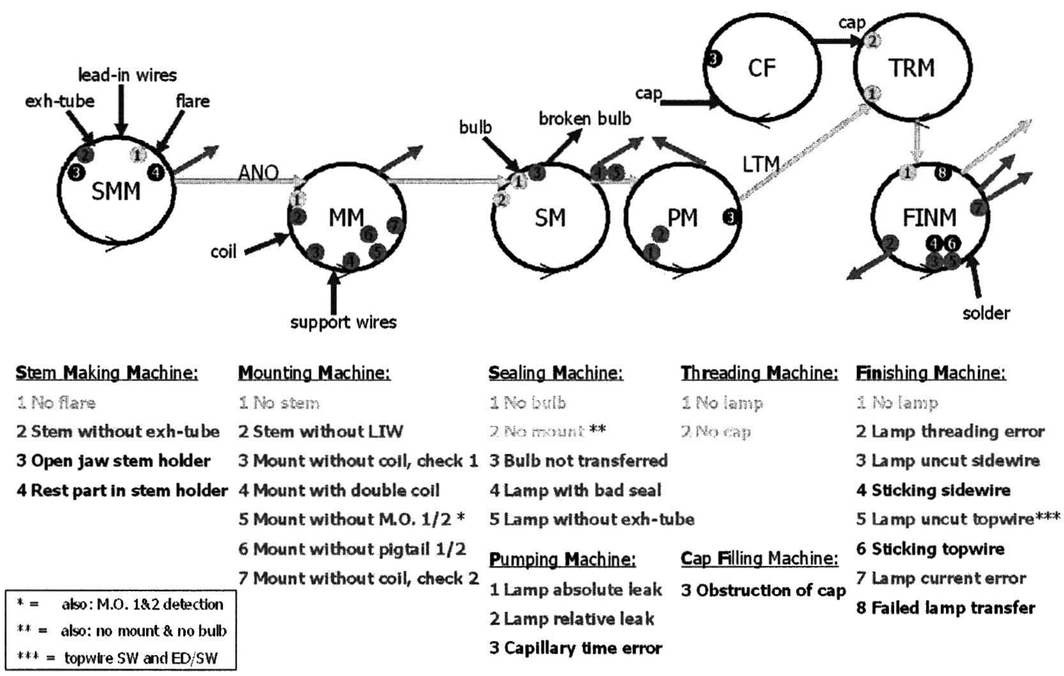

Figure 5.1: Overview of B-group production line

The semi-product that leaves this machine is called a stem. To reduce the internal pressures in the glass stem, a slow cool down process follows in the annealing oven (indicated as ANO). This open oven serves as a buffer at the same time.

5.2.2 Mounting machine

Stems then enter the mounting machine (MM) by a vacuum transfer and the presence of the lead-in-wires is checked. If one or two of the wires appear to be missing (or bent, which also results in this sensor giving a positive signal) the product will be thrown away and no coil needs to be added. An intelligent coil feeder system then provides coils on an indexing drum. A transfer unit picks up a coil and places it with both ends at the ends of the two lead-in-wires. Next, a mechanical movement clamps the ends of the wires around the coil ends to provide fixation.

The next sensor checks whether it is possible to send a current through the lead-in-wires and coil. Afterwards the lead-in-wires are spread. Then, two support wires are added by inserting them into the melted top of the stem.

The presence of these wires is checked by a sensor, and if present, the wires are folded around the middle of the coil, forming a pigtail, in order to support the coil. This is necessary because the coil elongates as it heats up during the use.

Before the semi-product leaves the mounting machine it is checked once again for the presence of a coil by sending a current through it once more. If the product is still conforming it moves on, if not it is rejected into a bin.

5.2.3 Sealing and pumping machine

The new semi-product, which is now called a mount, is then transferred to the sealing machine (SM) in which the bulbs are placed over the mounts and melted together. The infeed of both bulbs and mounts is checked. If the vacuum transfer of the bulb fails the bulb is either broken or has fallen out of the transfer unit. Because of this the corresponding product needs to be rejected.

After melting the mount and bulb together, the product is checked for its seal shape and the presence of the exhaust-tube (it might have been broken during transport, it is also possible that the entire lamp fell out of the machine during the transfer). All rejected products are again thrown into a reject-bin.

The newly formed lamps are then transferred, via vacuum transfer units, to the pumping machine (PM). Here the repetitive pumping process begins in which all air is gradually replaced by a neon-fluor gas mixture. To make this possible a pumping unit is attached via rubber tubes to the exhaust-tube of the lamp.

When this process is finished the lamp is checked for possible leaks (dependent on the actual pressure value a categorization is made between good lamps, relative leaky lamps and lamps with an absolute leak). Next, the exhaust-tube through which the gas has flown is pinched.

It is checked how long this pinching takes to make sure that the exhaust-tube that is broken in this process is of a high enough temperature to ensure easy breaking (otherwise the pinching blocks would be damaged by the hard glass). Leaky lamps and those in which the pinching was too fast or too slow will be removed.

5.2.4 Threading and finishing machine

The lamps then go on the lamp transport machine (LTM) which serves as the second buffer in the line. Lamps are transported towards the threading machine (TRM) in which a cap will be placed over the lead-in-wires at the bottom side of the lamp. This cap is first filled with cement in the cap filling machine (CF), in order to make it stick to the lamp.

After threading the lamp is transferred to the finishing machine (FINM) in which the result of the threading is checked by trying to send a current through the lead-in-wires and coil. If no current flows the threading has failed (for instance one of the lead-in-wires might be trapped inside the cap), or the product is not compliant in another way.

Good products proceed to the wire cutting process in which the two topwires (or the topwire and the sidewire, dependent on product type) are being cut. After transferral to the lower mill of the finishing machine the lamps arrive at the soldering unit which places a bit of solder on the ends of the wires to ensure connection to the cap.

The current through the product is then finally measured to assess whether the lamp is working or not. If the lamp appears to be leaking the current value will be different. Lamps through which the measured current flow is not within specifications are then finally rejected at the end of the line.

Good products leave the production line for automated transport to the packaging line.

5.3 Scope determination

It is not realistic nor sensible to choose the entire production line as the scope for the current experiments with this tool. The main objective is to assess the *potential* of the developed binary process quality control methodology. It is therefore preferred to customize and thoroughly test the newly developed tool for the beginning of the line only, rather than creating a partly operationalized tool for the entire line.

The current attention is therefore specifically restricted to the first two machines, the stem making machine and the mounting machine.

The training and testing of the developed tools is performed during two periods of one month each, on the B4 line of Philips Lighting located in Surabaya, Indonesia.

5.4 Analysis technique customization

Employing the analysis tool, that was developed in Section 4.3, consists out of a number of steps:

- calculate the entitlement level
- try to fit a theoretical distribution
- perform the entitlement check for new data

To provide a clear exemplary procedure for employing the newly developed process quality control methodology, each of these steps will be elaborately discussed.

5.4.1 Entitlement calculation

The entitlement calculation aims at retrieving an average value for the amount of product blemishes for those cases in which only problems with normal causes are present in the production process. This procedure requires quite some effort as illustrated in Section 4.3.1:

- identify what is counted by the binary sensors in the process under consideration
- identify the type of area of opportunity and find the proper basis for it
- identify abnormal production periods by using information from the control system and documentation
- employ the developed automated entitlement calculation procedure
- check whether datapoints that are characterized as abnormal indeed display assignable causes, and whether normal datapoints lack assignable causes

As can be concluded from the description of the production process of incandescent lamps at Philips Lighting, the binary sensors check product compliance. If a sensor detects a blemish for a certain product, the entire product is classified as nonconforming. The product is then rejected, without being tested on other characteristics by the other sensors. The count data per sensor therefore consists out of rejected products by that sensor.

As nonconforming products are not tested by other sensors anymore, the amount of possible detected blemishes per product is strictly limited to one. The amount of lamps that can be nonconforming therefore only depends upon the amount of lamps produced. The area of opportunity in the current case is therefore formed by the amount of lamps produced in a certain production period.

Selecting a proper time basis for *one production period* involves several considerations. On the one hand side, the shorter this time basis, the sooner the analysis tool can display the information it retrieves. After a quick diagnosis, this improves the possibilities for useful feedback on the found root-causes. On the other hand, the time basis needs to be long enough to have a substantial amount of rejects in each period (averages around 0, 1 or 2 hamper the entitlement calculation). After trying several time bases "*one hour*" proved to be suitable for the current case

Next, abnormal production periods need to be identified. Those production periods that were not intended to produce lamps but were used for minor or major adjustments, investigations, change-overs or maintenance activities are identified and excluded from the dataset.

This identification is based on the amount of lamps that have been produced p_{m_t}, with m the relevant machine of the production line. The threshold has been based on $\frac{1}{2}$ the capacity of the machines, which corresponds to approximately $p_{m_{t_{threshold}}} = 2000$ lamps. This information has been retrieved, real-time, from the control system of the production line. A verification is performed by consulting the documentation of production and maintenance activities and planning.

The next important step is the employment of the automated entitlement calculation procedure. The needed information is the amount of blemishes and the corresponding area of opportunity. For the current case the former values are represented by the amount of rejects of each of the sensors per hour x_{i_t}. The latter values are formed by the amount of lamps produced per hour by the machine on which the respective sensor is located p_{m_t}. As can be concluded from the description of the monitoring equipment at Philips Lighting as described in Section 5.2, the following information is available:

x_{1_t} = amount of *stem without exhaust-tube* rejects

x_{2_t} = amount of *stem without lead-in-wire* rejects

x_{3_t} = amount of *mount without coil, check 1* rejects

x_{4_t} = amount of *mount without support wire 1* rejects

x_{5_t} = amount of *mount without support wire 2* rejects

x_{6_t} = amount of *mount without support wire 1 & 2* rejects

x_{7_t} = amount of *mount without coil, check 2* rejects

p_{SMM_t} = amount of lamps produced on the stem making machine

p_{MM_t} = amount of lamps produced on the mounting machine

All this data is available *for every hour.* In the training period of one month, which was the basis for the entitlement calculation, 275 production hours have been analyzed. In total, 33 production periods were characterized as not strictly intended to produce lamps. For the other 242 production hours, every combination of sensor value and amount of lamps produced is retrieved from the control system of the incandescent lamps production line. This is performed with the ActiveFactory application of the software package Excel, on an automated and real-time basis. The 3388 datapoints are transformed into 1694 reject percentages per sensor per hour, in the following manner:

$$r_{i_t} = \frac{x_{i_t}}{p_{m_t}} \cdot 100\% \tag{5.1}$$

where:

r_{i_t} = reject percentage for sensor i on machine m

x_{i_t} = amount of counted rejects for sensor i

p_{m_t} = amount of products produced on machine m

Next, the type of the data needs to be assessed. No subgroups can be distinguished within the data. The datapoints are generated over time though, with one datapoint for every hour. The data therefore needs to be treated as time series data. The upper limit is therefore calculated with equations 4.4 and 4.5, as illustrated for time series data in Section 4.3.1.

The results of the subsequent automated entitlement calculation procedure are summarized in Table 5.1. The overall average of normal and abnormal production periods is indicated. Furthermore the final entitlement level that was found, and the upper limit belonging to the last filtering cycle are presented.

sensor:	1	2	3	4	5	6	7
$\bar{r}_i$	0.25	2.5	2.1	0.30	0.14	0.18	0.36
r_{ENT_i}	0.21	2.3	1.5	0.20	0.11	0.17	0.30
r_{UL_i}	0.68	4.2	3.6	0.70	0.44	0.47	0.85

Table 5.1: Results of automated entitlement calculation. Final entitlement level and upper limit are indicated for counts per unit of area of opportunity

The final step in the entitlement calculation procedure is the analysis of convergence behavior and assignable causes. In the table below the amount of abnormal values detected during the entitlement calculation for each sensor is presented. The convergence behavior is illustrated by indicating the amount of values filtered in each of the filtering cycles.

sensor:	1	2	3	4	5	6	7
# abnormal values	11	13	21	15	14	9	15
1^{st} cycle	7	11	16	10	11	6	8
2^{nd} cycle	4	1	4	5	3	2	4
3^{rd} cycle	0	1	1	0	0	1	2
4^{th} cycle	0	0	0	0	0	0	1
5^{th} cycle	0	0	0	0	0	0	0

Table 5.2: Convergence behavior of entitlement calculation. The amount of abnormal values is indicated, as well as in which filtering cycle the points were identified as abnormal and thus excluded from the dataset

For periods in which no abnormal values were detected, no causes appeared to dominate on the first two machines. It can therefore be affirmed that normal production periods did not contain assignable causes.

The 98 abnormal values occurred in 75 distinct production periods. This means that in approximately 30% of the production hours problems were present on the first two machines of the incandescent lamps line. Judging from the frequency of occurring problems, while being present at the production line for a period of 3 months, this seems a very realistic figure.

5.4.2 Fitting a distribution

In Section 4.3.2 the steps involved in finding a theoretical distribution fitting practical data were discussed. The current challenge is to find out whether a well fitting distribution exists to the normal datapoints that have been found by the entitlement calculation process. These steps [Law01] will now be employed and the results for one of the datasets (the *mount without coil check 1* - sensor) will be briefly discussed.

Assess sample independence

As a measure of sample dependence the correlation between subsequent datapoints is calculated. This correlation $\rho_{t,t+1}$ between r_t and r_{t+1} for $t = 1, ..., n_t - 1$ appears to be $\rho_{t,t+1} = 0.56$. This is substantially higher than 0, therefore the data from subsequent time periods are not independent.

The scatter diagram of Figure 5.2 leads to the same conclusion. Here, the $\frac{1}{2}n_t$ pairs consisting of r_t and r_{t+1} for $t = 1, ..., n_t - 1$ are plotted. If the r_t's would be independent these values would be scattered randomly throughout the first quadrant. A clear positive correlation appears to exist, as indicated by the least squares estimate line that has a positive slope.

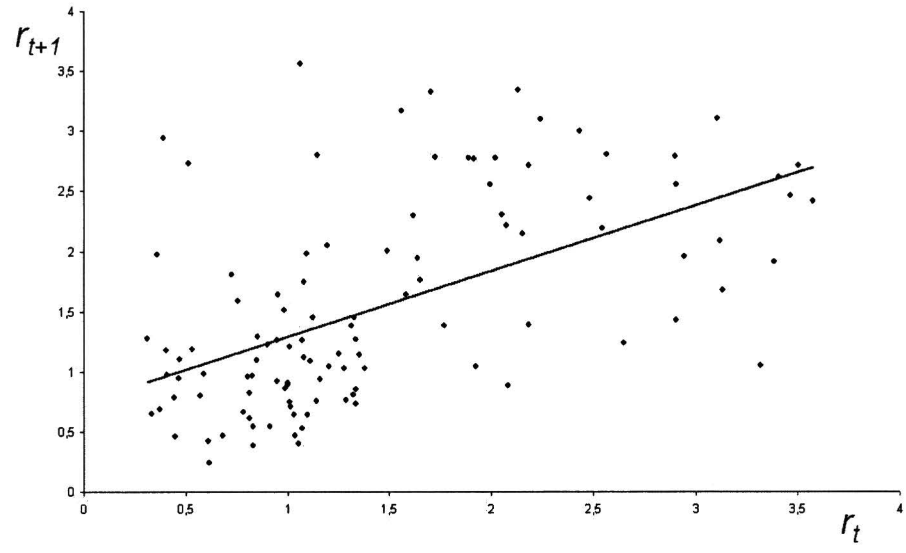

Figure 5.2: Scatter plot of the pairs of r_t and r_{t+1} showing a positive correlation

The figure does indicate however that the correlation is especially apparent for those time periods with higher reject percentages (the upper right zone of the first quadrant). As problems apparently tend to stick around period after period, this indicates that the problems in these relatively troublesome time periods are not dealt with effectively. This coincides with the expectation of the situation, and the positive correlation is therefore not a surprise either. The implementation of the process quality control methodology that is currently under development could vastly improve this situation by pinpointing the present root-causes of these troubles.

Below the scatter diagram of the zone of datapoints for which $r_t \in [0, 1.5]$ is presented. The correlation coefficient is still significant $\rho_{t,t+1} = 0.34$. The exponential, Binomial and Poisson distribution that require independent data (memoryless property) are therefore not suitable for this dataset.

The datasets of the other sensors are not independent either, as shown in Table 5.3. The correlation $\rho_{t,t+1}$ between r_t and r_{t+1} for $t = 1, ..., n_t - 1$ is shown.

sensor:	1	2	3	4	5	6	7
$\rho_{t,t+1}$	0.15	0.32	0.56	0.43	0.33	0.36	0.15

Table 5.3: Correlation $\rho_{t,t+1}$ between subsequent datapoints for all 7 sensor - datasets

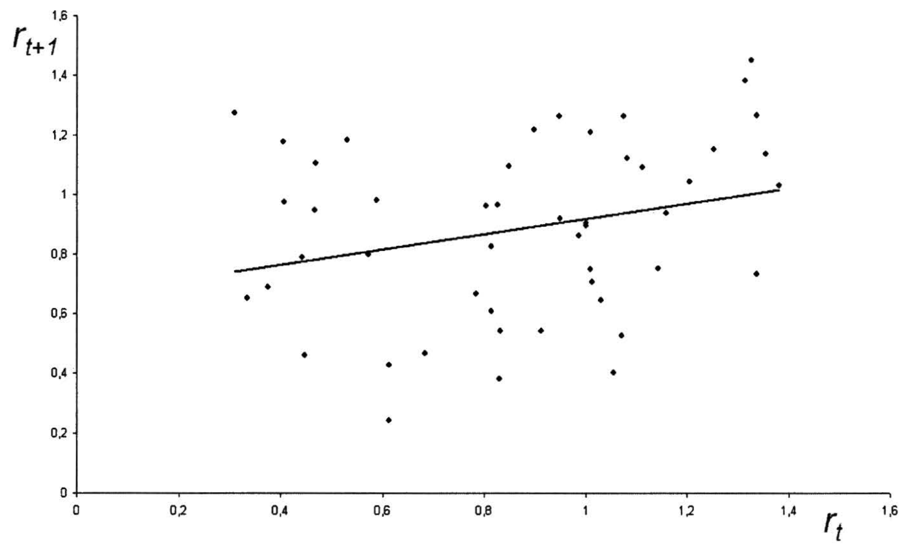

Figure 5.3: Scatter plot for $r_t \in [0, 1.5]$, still showing a positive correlation

Hypothesize families of distributions

No samples within the data can be distinguished between which, as opposed to within which, variation would logically occur. Therefore, the binomial distribution appears to be unsuitable once more.

The coefficient of variation of the dataset is $cv = 0.57$. For exponential and Poisson distributions $cv = 1$, regardless of the shape parameter β or λ. Since $cv = 0.57$ is significantly different from 1, the use of exponential and Poisson distributions is ruled out once more as well.

As the discrete count levels x_t are divided by the respective amount of lamps produced p_{m_t} to make them comparable, the resulting percentages r_t are continuous figures (see Section 5.4.1). For the continuous gamma and Weibull distributions a $cv < 1$ means that the shape parameter $\alpha > 1$.

In Table 5.4 the cv values for the other datasets are presented. Most sensor - datasets have a similar cv to the *mount without coil check 1* dataset. There are only two exceptions. The very low value of $cv_2 = 0.27$ belonging to the *stem without lead-in-wire* sensor indicates that the σ of this dataset is very low compared to its mean. The $cv_5 = 1.02$ of the *mount without support wire 1&2* check indicates that μ and σ of this dataset are almost equal.

sensor:	1	2	3	4	5	6	7
cv	0.74	0.27	0.57	0.75	1.02	0.65	0.65

Table 5.4: Coefficient of variation cv for all 7 sensor - datasets

Below a histogram of the data r_t for $t = 1, ..., n_t - 1$ is presented. Hereto, the reject percentages are divided into bins with a width of 0.30 and the respective frequencies are counted.

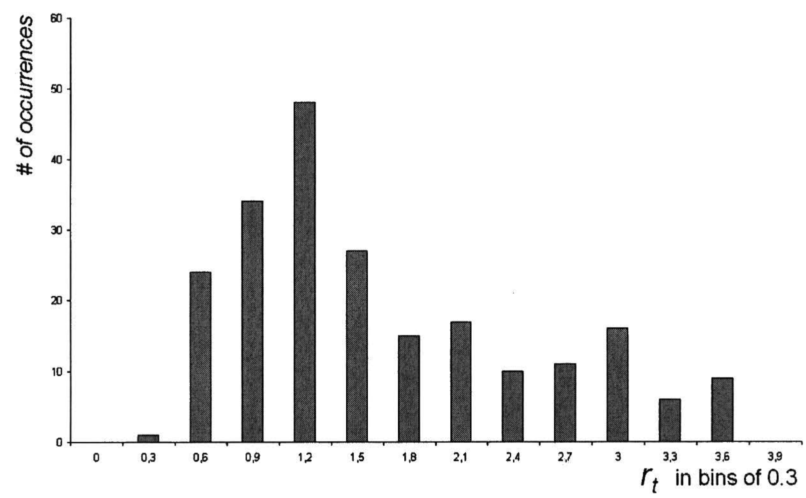

Figure 5.4: Histogram of number of occurrences of r_t in bins

The gamma and Weibull families of distributions seem most likely to provide a good fit to the practical dataset.

Estimation of parameters

The ultimate form of the gamma and Weibull distributions is dependent upon a shape and a scale parameter, α and β respectively. The Maximum Likelihood Estimator (MLE) algorithm in Matlab is used to determine the values for α and β for both distributions:

$$\text{Gamma}(\alpha, \beta) = (3.13, 0.48)$$

$$\text{Weibull}(\alpha, \beta) = (0.37, 1.88)$$

The latter result $\alpha < 1$ (!), indicates that the best possible fit of the dataset by a Weibull function is monotonously decreasing, which is not similar to the histogram of the dataset at all. Weibull is therefore not able to provide a good fit to the dataset under consideration.

The same situation is encountered for the other datasets. For these sets the gamma distribution therefore appears most suitable as well, with exception of the 5^{th} dataset (for which $cv \approx 1$). The Maximum Likelihood Estimators with 95% confidence intervals for the parameters of the gamma distribution are tabulated below for all datasets.

sensor:	1	2	3	4	5	6	7
α	**2.2**	**14**	**3.1**	**2.0**	**1.7**	**2.6**	**2.2**
$\alpha_{lowerbound}$	1.7	11	2.4	1.6	1.2	2.1	1.8
$\alpha_{upperbound}$	2.7	17	3.9	2.5	2.2	3.2	2.6
β	**0.098**	**0.16**	**0.48**	**0.10**	**0.077**	**0.066**	**0.14**
$\beta_{lowerbound}$	0.076	0.13	0.36	0.079	0.055	0.050	0.11
$\beta_{upperbound}$	0.12	0.20	0.60	0.12	0.099	0.082	0.17

Table 5.5: Estimates and 95% confidence interval bounds for the shape and scale parameter α and β for fitting a gamma distribution to all 7 sensor - datasets

Determining quality of fit

In Figure 5.5 the plot of the gamma$(3.13, 0.48)$ distribution in the histogram of the practical dataset is shown to assess its conformance.

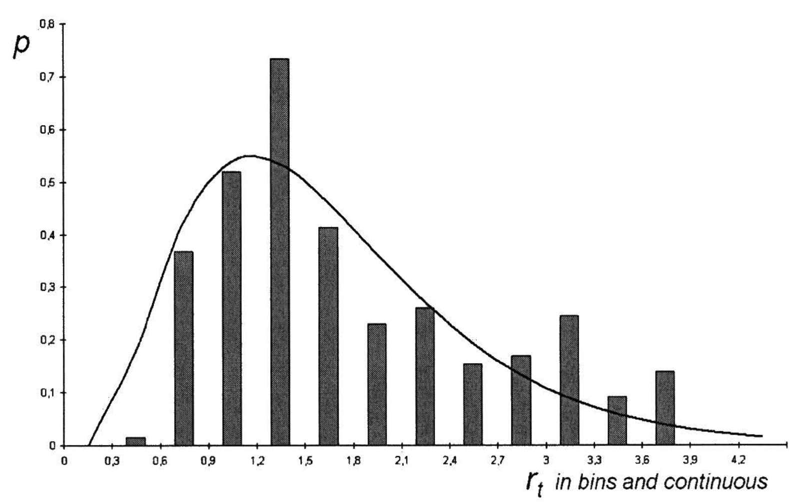

Figure 5.5: Approximation of practical dataset by gamma$(3.13, 0.48)$ distribution

The differences appear to be too substantial to accept the distribution. The Kolmogorov-Smirnov tests reach the same conclusion for all datasets.

As $cv \approx 1$ for the 5^{th} dataset, a Poisson distribution may be more appropriate for this sensor. The MLE for λ appears to be 0.166. The discrepancy between the Poisson$(\lambda = 0.166)$ distribution and the histogram of the 5^{th} dataset appears to be too large as well.

5.4.3 Entitlement check

For the sensor data no well fitting theoretical distribution has been found. This is partly caused by the dependency of the data. Furthermore, the amount of variation in the data is high, which would decrease if more datapoints were obtained.

As a result, the entitlement check of the data in the training period needs to be performed with the use of equation 4.19, based on the descriptive statistics that were found to represent the normal causes - distribution. As explained in Section 4.3.4, these estimators coincide with the statistics that were used in the last filtering cycle of the entitlement calculation.

The same 98 values that were characterized as abnormal during the entitlement calculation are therefore detected by the entitlement check as well. These abnormal values occurred in 75 distinct production periods. More precisely, 58 periods of the 242 were characterized by 1 abnormal value. Another 11 periods were characterized by 2 simultaneous abnormal values. Finally, during 6 production periods 3 sensor values were abnormal.

In 35 of the 75 abnormal production periods an assignable cause has been detected. The conclusion therefore is that only for approximately 1 out of every 2 abnormal cases an assignable cause could be found. Given the complex nature of the search process for assignable causes it is very likely that certain causes have been present for most of the other abnormal cases as well, but could simply not be detected. One reason is the difficulty of employing the camera tool in the right time and right manner. Furthermore, Indonesian employees sometimes tampered with the process without explaining present root-causes. In these situations conclusions on the presence of assignable causes could not be drawn with full confidence.

The trade-off in these instances is that inclusion of these periods as training cases increases the amount of training of the tool, while decreasing the confidence in classification correctness of these training cases. The choice has been to assign root-causes to production periods only in those cases in which it is very sure that this root-cause had been present. This choice is made because avoiding misclassification of training cases improves the future classification correctness rate, which is the ultimate goal of the methodology.

Taking these difficulties into account the overall resulting entitlement of production periods as normal or abnormal seems to be sensible.

5.5 Classification technique customization

It is the objective of the classification tool to use the information as delivered by the analysis tool, to retrieve and indicate the present root-causes. Hereto, the binary features delivered by the analysis tool are put into a state vector that describes the system state of the production line, as discussed in Section 4.4.1. As the retrieval of root-causes is quite a complicated task, it is important to use the delivered information optimally. A smart choice of binary features can ensure this.

5.5.1 Binary feature selection

The three types of phenomena that can be delivered by the analysis tool were discussed to be (see Section 4.4.1):

- abnormal values for a certain sensor
- special correlations between abnormal values if relevant from an engineering point of view
- events like stops, breakdowns, change overs, etc.

The first category of phenomena can be directly translated into binary features.

Regarding the second category, the following correlations could be very interesting from an engineering point of view:

- abnormal stem without lead-in-wires and abnormal mount without coil, check 1; because the coil is placed in between the lead-in-wires.
- abnormal mount without support wire 1; and normal mount without support wire 1 & 2; because then the insertion or supply of support wire 1 is failing sometimes, rather than a problem with rolling which would affect both wires at the same time.
- abnormal mount without support wire 2; and normal mount without support wire 1 & 2; because then the insertion or supply of support wire 2 is failing sometimes, rather than a problem with rolling which would affect both wires at the same time.
- abnormal mount without support wire 1 & 2; and abnormal mount without coil, check 2; because this could mean that the coil is somehow damaged after the 1st check, because of a wrong rolling process for the support wires

So, in total, there are 11 binary features for describing the state of this part of the production line. The system state vector is indicated in Figure 5.6.

$$
s_t = \begin{bmatrix} bf_{1_t} \\ bf_{2_t} \\ bf_{3_t} \\ bf_{4_t} \\ bf_{5_t} \\ bf_{6_t} \\ bf_{7_t} \\ bf_{8_t} \\ bf_{9_t} \\ bf_{10_t} \\ bf_{11_t} \end{bmatrix} = \begin{bmatrix} \text{abnormal } \textit{stem without exhaust-tube} \text{ value} \\ \text{abnormal } \textit{stem without lead-in-wires} \text{ value} \\ \text{abnormal } \textit{mount without coil check 1} \text{ value} \\ \text{abnormal } \textit{mount without support wire 1} \text{ value} \\ \text{abnormal } \textit{mount without support wire 2} \text{ value} \\ \text{abnormal } \textit{mount without support wire 1\&2} \text{ value} \\ \text{abnormal } \textit{mount without coil check 2} \text{ value} \\ \text{both } \textit{stem without LIW} \text{ and } \textit{coil check 1} \text{ high} \\ \text{both } \textit{support wire 1\&2} \text{ and } \textit{coil check 2} \text{ high} \\ \text{high } \textit{support wire 1} \text{ but low } \textit{support wire 1\&2} \\ \text{high } \textit{support wire 2} \text{ but low } \textit{support wire 1\&2} \end{bmatrix}
$$

Figure 5.6: System state vector that describes the current status of the first part of the production line

The third category of phenomena are not included yet. The possible benefits of including the occurrence of these kind of events will be investigated in Section 5.6.1.

5.5.2 Training the tool

Training the developed binary methodology consists of linking the occurring root-causes to their state vectors. To make this possible all root-causes of problems that occurred during the training period need to be known. As discussed in Section 4.2, the use of product images as in the spatial signature analysis technique may be very useful to find the root-cause of a problem. Pictures of the semi-products should therefore be taken at those instances that a problem occurs. Simply photographing all semi-products would lead to the need to diagnose 4000 different photos every hour (the machine capacity) though. Obviously this is too large a training effort. For the incandescent lamps production line typically only a small percentage of semi-products is affected by a problem however. To avoid these high training burdens a camera tool should be developed that is able to take pictures of rejects only, as these pictures would reveal most about the root-cause of a problem.

The control system of the production line, S7, controls the rejection of the products. If a sensor rejects a product it sends a signal to the control system. The control system then marks the index position of this product in the machine mill. It is therefore able to trigger the camera to take a picture of exactly these rejects.

In this way the camera tool would be able to take pictures of rejects during the operation and of the resulting nonconforming product. In order to make semi-products that go into the operation diagnosable as well the functioning need to be adapted. The camera should take pictures of all products, but only save them if the semi-product appears to be nonconforming. This is the ultimate way of functioning of the camera tool. As a result it can be placed up to 100 positions before or after the sensor position. Therefore, it is possible to investigate the quality characteristics of the semi-products at 200 different positions.

An overview of the connection between the control system and the camera tool, and the pulses used for triggering can be found in Appendix A.

The developed camera tool is now used to investigate the root-causes of problems that occur in the production line. To know when problems are occurring, the real-time overview of abnormal sensor values is used as a trigger. Furthermore, intensive communication with the operating staff of the line is needed. The experimental setup with this camera is illustrated in Figure 5.7 and schematically in Figure 5.8.

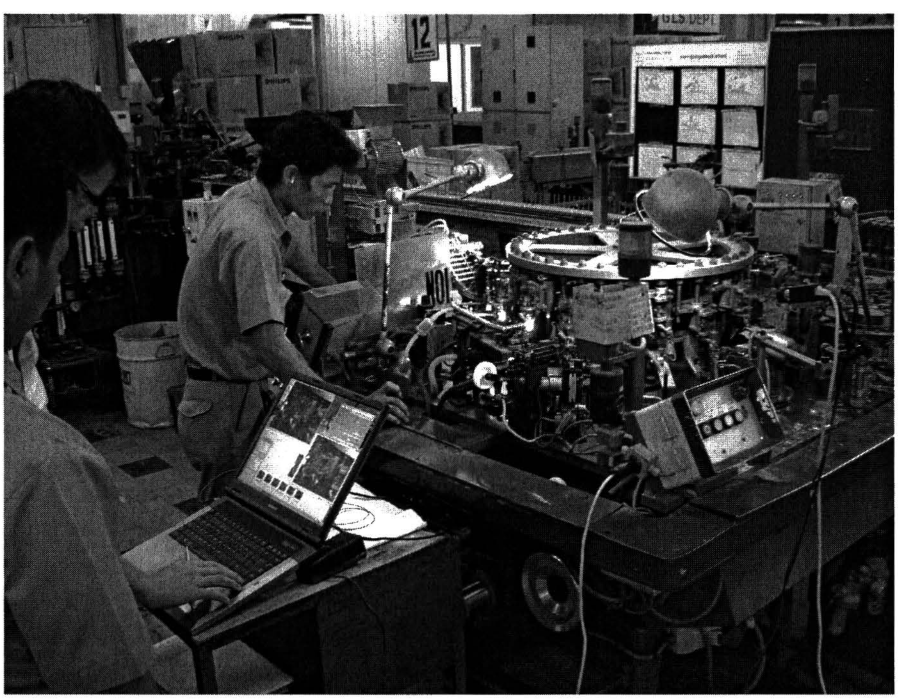

Figure 5.7: Overview of experimental setup. The camera is located on the right hand side of the mounting machine for a check on the rolling process

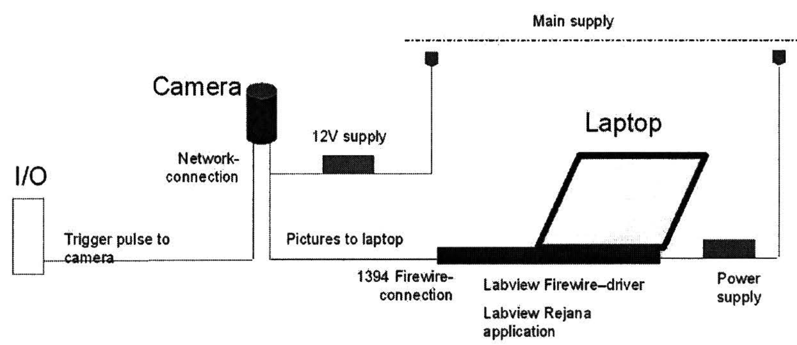

Figure 5.8: Schematic overview of the experimental setup

The 242 production hours intended for producing lamps during the training month, have been used for the training of the tool. As discussed in Section 5.4.3, the entitlement check characterized 75 production periods as abnormal. During these periods, the interesting problems on the first machines of the pilot line have been diagnosed with the camera tool.

The following root-causes of problems appeared to dominate:

1. Lead-in-wires are being bent during vacuum transfer from annealing oven to mounting machine (and they are therefore not detected, leading to a *stem without LIW* failure), also this could result in the failure to place a coil in between the lead-in-wires, leading to a failed *coil check 1.*
2. Lead-in-wires broken during bending process. Because of bad quality semi-product or misaligned bending unit the lead-in-wires break after the *stem without LIW* check which leads to a failed *coil check 1.*
3. Coil misaligned on drum because of bad setting of the drum alignment plate. Coil transfer fails, or coil clamped at one side of the stem only. The stem will fail *coil check 1.*
4. Bad coil transfer system. The system can be misaligned, or the sucker settings are wrong. Coil transfer fails, leading to a failed *coil check 1.*
5. Rolling process for the support wires is misaligned which causes failure in the rolling process (and the support wires are then not detected, leading to a failed *mount without support wire 1&2 check*), also this can damage the coil which would cause the mount to fail on *coil check 2.*
6. Insertion or supply of support wire 1 malfunctioning.
7. Insertion or supply of support wire 2 malfunctioning.

Each of these root-causes reoccurred a number of times. In total 35 times one of these assignable root-causes was found. Every time one of these root-causes occurred the corresponding state vector was retrieved from the analysis tool.

The training cases that belong to the root-causes 2, 3 and 4 appear to have exactly the same state vector. This means that these root-causes are indistinguishable with the present sensors. Therefore, they are combined into one single root-cause class that is distinguishable from all other root-causes. This yields the following root-causes classes:

rc_1 = bent lead-in-wires

rc_2 = misaligned coil on drum, bad coil transfer system or lead-in-wires broken

rc_3 = misaligned support wire rolling process

rc_4 = malfunctioning insertion or supply of support wire 1

rc_5 = malfunctioning insertion or supply of support wire 2

rc_6 = no root-causes present

In Appendix B an overview is presented of the state vectors of all training cases belonging to each of the root-cause classes. Also the state vectors of instances in which no root-causes were found to be present are presented.

It can be seen that 3 of the 21 distinct state vectors have occurred in different situations. That is, each of these three vectors have been found to occur while a certain root-cause was present (a different root-cause for each of these vectors !), but also while no assignable causes have been found. As discussed earlier, due to the complexity of finding root-causes, the fact that no root-causes could be found does not necessarily mean that no causes have been present for these cases. Furthermore, as it is preferred to develop a sensitive tool, it is less harmful for the tool to suggest the possible presence of root-causes while in fact none are present, than to fail to indicate actual presence. Inclusion of the double occurring state vectors as training cases for both the respective *root-cause* classes and the *no problems present* - class leads to the algorithm indicating both options. The choice is to omit these 3 state vectors as training cases for the *no problems present* - class.

Following the same sensitivity argument, 3 rarely occurring (only once or twice during one month) other *no problems present* - state vectors have been omitted as training cases for this class. It is again preferred to increase the chances of coupling situations with alarms to their root-causes.

The state vectors that were found to belong to the different occurring root-causes are put in the database of the comparison algorithm. This algorithm is programmed in Matlab. The m-file can be found in Appendix C.

5.6 Testing the customized methodology

During the test period of one month, 270 production hours were analyzed by the analysis tool. Of these production periods, 42 were judged as not strictly intended to produce lamps. Therefore, 228 datapoints were available for testing. Of these datapoints, the entitlement check characterized 73 as abnormal.

Next, the state vectors of these 228 production periods were constructed by using the binary features as delivered by the analysis tool. The diagnosis tool then compared these state vectors to the vectors of the training cases, to yield an expectation of the present root-cause.

As a check for classification correctness, the camera tool was used for the abnormal cases and for a sample of the normal cases as well. For 38 of the 73 abnormal cases a root-cause could clearly be distinguished by the camera tool, whereas a root-cause could not be found for the remaining 35 cases.

Summarizing, the following different situations can be distinguished:

1. no abnormal values, no assignable cause present
2. certain abnormal values, no assignable cause detected
3. no abnormal values, a certain assignable cause is present
4. certain abnormal values, a certain assignable cause is detected

The **first category** of cases occurred 155 times. Therefore, the classification of this category of cases has been checked with a sample, of 10 cases. None of the 10 investigated sample cases without abnormal values, were found to possess any root-causes. Therefore, this category is concluded to possess a 100% correct classification rate. This is a result of the high sensitivity of the analysis tool.

The **second category** occurs more often as a direct consequence. Although the entitlement check indicates the abnormality of the resulting level and suggests the presence of an assignable cause, research with the camera tool did not result in finding such a distinguishable cause. As indicated earlier, the investigation with the camera tool may have been employed in the wrong place, time or way. Some problems simply seize before they can be detected. Alternatively, rarely occurring root-causes may have been present that are not included in the list of dominating occurring root-causes (see Section 5.5.2), and are therefore not subject to investigation.

For the test period this situation occurred 35 times. Of these 35 times the diagnosis tool classified 21 cases as not possessing any of the dominating root-causes. Therefore, 60 % of this type of situations can be said to be correctly categorized. The remaining 14 cases, equivalent to 19% of all abnormal cases, are somehow divided between cases in which the employment of the camera tool failed to detect the present root-cause, and in misclassified cases because of oversensitivity.

The **third category** can not occur because this would violate the employed definitions. Root-causes do occur in periods without abnormal values but are sometimes effectively dealt with on a very short notice. These cases therefore do not result in abnormal values, and are therefore said to be normal, so no assignable causes are said to be present. Whether or not this is a reasonable assumption depends upon the specific situation. For Philips Lighting it was concluded at the end of Section 5.4.1 that no causes seemed to dominate in cases without abnormal values.

The **fourth case** is by far the most interesting. Here, root-causes are present, unlike in the first two categories. Furthermore, these root-causes are not already dealt with effectively, as opposed to what happens in the third category. So this is the situation for which the process quality control methodology has been designed. The diagnosis tool should indicate the right root-cause, which would allow the subsequent feedback by the operators.

Two situations are possible. Either the right root-cause is indicated, which is a correct classification, or the wrong root-cause is indicated, a misclassification.

As stated, during the testing period of one month, use of the camera tool concluded that 38 production periods that were characterized as abnormal were found to belong to one of the root-cause classes. The amount of test cases for each root-cause class is indicated below. Furthermore, the correct classification rate for each root-cause class is presented. Finally, for the misclassified test cases it is indicated to which other root-cause classes the cases have been classified.

root-cause class:	rc_1	rc_2	rc_3	rc_4	rc_5
# test cases	9	12	8	5	4
correct classification rate	89%	92%	100%	80%	100%
misclassification to	rc_2	rc_4	-	rc_1	-

Table 5.6: Classification results

5.6.1 Classification rate optimization

Usefulness of current features

Regarding the first category of features, the abnormal sensor values, all features appear to contribute except the *stem without exhaust-tube* feature. As all training cases with this feature were found to possess none of the currently considered root-causes it is likely that a certain root-cause on the stem making machine should be taken into account as another dominating root-cause. It is expected that bent or clogged channels of the input carroussel of exhaust-tubes, leads to increased failure on this check. Future investigation on this matter is therefore recommended.

Earlier, it has been assumed that the classification rate would improve by taking correlations that are relevant from an engineering point of view explicitly into account. This assumption was based upon the way of working of the comparison algorithm. As this algorithm looks at the *amount* of binary features that are the same for the training and testing case (in a weighted manner) it is very likely that it is beneficial to include those correlations which point towards specific root-causes.

For the current scope of the tool the tests can partly affirm this assumption. For root-cause class 5 (rc_5) the classification correctness rate is increased by 20% because of the inclusion of bf_{11}. For the other root-cause classes the inclusion of the correlations does not lead to improvements yet.

The reason lies within the limited scope of the current application of the tool. Currently only the first two machines are considered, which means that only 7 sensors deliver their input at the moment. Therefore, for the training cases only 21 different state vectors have been found, of which only 14 described system states with a root-cause present. Most test cases are therefore exactly similar to one of these training cases (76% to be precise). Logically, the inclusion of correlations can only be beneficial for cases that are not exactly similar to training cases, which is rare at the moment.

If the scope of the developed methodology will be extended to the entire production line, information from 22 sensors will be used. Because of more simultaneously occurring abnormal values in different parts of the line, many more cases will have vectors that are not exactly similar to training vectors. It is expected that the correlations between abnormal values, that are relevant from an engineering point of view, will then further improve the classification correctness rate.

Additional binary features

It is possible to extend the delivered information with all kinds of events, like stops, breakdowns, change overs, shift identification tags and maintenance activities.

The training period indicated that no distinction in the state vectors of the dominating root-causes 2, 3 and 4 existed. Typically the problems with settings of the drum alignment plate and the coil transfer system (root-causes 3 and 4) are expected to occur more frequently after change overs of coil type. Root-cause 2, broken lead-in-wires, is expected to be unaffected by the coil type change over. As an exemplary test, the system state vector has therefore been extended with this binary feature:

$$bf_{12_t} = \text{change over of coil type in period } t-1$$

Of the 8 (similar) training state vectors belonging to the combined root-cause class rc_2, 2 vectors correspond to dominating root-cause 2, whereas 6 vectors corresponded to dominating root-causes 3 and 4. None of the root-cause 2 vectors appeared to have occurred immediately after a change over, whereas 3 out of the 6 other vectors did occur after a change over of coil type.

Inclusion of the additional binary feature leads to the correct classification of 4 of the 9 occurrences of dominating root-causes 3 and 4 during the test period. The other 5 occurrences of these dominating root-causes did not follow a change over and can therefore not be distinguished by the new binary feature. Nevertheless, 44 % of the test cases belonging to these dominating root-causes can now, correctly, be further identified as belonging to these specific root-causes in stead of to the broader category of rc_2.

5.7 Summary

First, the process of producing incandescent lamps was elaborately discussed, and attention was devoted to the binary sensors involved. Next, the needed customization steps of the analysis tool were discussed, among which the determination of a suitable area of opportunity and the execution of the developed automated entitlement calculation. The validity of the resulting characterization of datapoints was checked with a camera tool and discussed. Finally, it was tried to characterize the datasets with theoretical distributions but the Kolmogorov-Smirnov tests indicated substantial discrepancies. Therefore, the final entitlement check of the training data was based on the descriptive statistics of the data.

The first step in the customization of the classification technique was the selection of relevant binary features. Based on engineering knowledge some interesting correlations were taken into account to characterize the system state. Subsequently, the development of a camera tool that knows which semi-products will be rejected was discussed. With help of this camera the present root-cause can be identified, by diagnosing pictures taken of rejected products before, during or after a faulty operation. The dominating root-causes for the training period were retrieved in this way, and linked to the system state vectors of the periods in which they occurred.

The next step has been the testing of the newly developed methodology by assessing its classification correctness rate for a testing period of one month. The resulting rates were satisfactory. Finally, some adaptations to further improve the performance of the tool were executed and tested.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

The objective of this research project was twofold. The first part was an exploratory study into the possibilities of developing a binary process quality control methodology. The second part consisted out of the implementation of the tools that were developed. For the production process of incandescent lamps at Philips Lighting the methodology was customized and elaborately tested.

Binary process quality control methodology

As binary sensors in general offer less informative data the application of analysis techniques to describe the system state appeared limited. As a consequence, direct classification techniques based on modeling of the input-output relationship were impossible for the practical case under consideration. To be able to indicate root-causes, the classification technique therefore needed to be built on a training procedure. In the search for a sound description of the system state, an entitlement calculation procedure was developed that can indicate whether each sensor value of a production period is normal or abnormal. By diagnosing system states with a developed camera tool a basis for comparison of cases was created.

Philips Lighting case

The results of the implementation of the methodology after customization at Philips Lighting were promising. Currently, the scope is limited to the first two machines of the line. Based on the information of the 7 sensors present on these machines, new cases could reliably be classified into 6 distinct root-cause classes. Classification correctness rates varied between 80% and 100%.

Studies on the reject data indicated that at the moment problems are not always dealt with effectively. The high positive correlation between paired hourly values for instance, indicates that problems tend to stick around. Informing the operators, technicians and engineers on the root-cause predictions of the developed process quality control methodology would give them important input for feedback into the process. It is expected that manufacturing performance will improve as a consequence. Research within Philips Lighting [Ree05] indicates that approximately 90k euro is wasted per line per year. As more than 200 lines exist all over the world, only small percentages of improvements could lead to substantial savings. However, it is not advised to implement the system yet. First, extension towards the entire line and additional investigation are recommended.

Other industries

The positive results for this practical case indicate that the developed methodology may be very useful for other binary industries as well.

6.2 Recommendations

Binary process quality control methodology

For the classification tool different algorithms with different ways of weighing the binary features have been tried on the current dataset. The, relatively simple, algorithm of Section 4.4.2 displayed the highest classification correctness for the current case, but for other situations or after more training or inclusion of more sensors more advanced weighing methods may provide better results. In Appendix E two alternative algorithms which are based on the selectivity of the features are included, as programmed in Matlab. It is recommended to compare the classification results of these different algorithms for new applications.

In Section 5.6.1 it was concluded that the domain of root-cause classes probably misses a root-cause on the stem making machine. It is recommended to investigate the inclusion of this root-cause. Some suggestions for possible causes were given.

Philips Lighting case

For Philips Lighting it is furthermore recommended, as mentioned, to extend the training of the developed methodology to the entire production line. If the other 5 machines will be taken into account as well, 15 more sensor signals will be available for describing the system state. Then, many more root-causes, that manifest on these other machines, are expected to become predictable. As the root-cause finding scope will increase, it is strongly recommended to perform thorough investigation into optimization of the usefulness of the distinct root-cause classes before implementing the methodology.

An interesting solution to increase the sensitivity of the tool regarding specific root-causes is the installation of extra sensors. Additional sensors, at smart places, can deliver very informative features about the system state. It is strongly recommended to investigate the extension of the monitoring tool for improvement of classification correctness and an increase of classification accuracy within root-cause classes.

Other industries

The recommendation for other binary industries is to test the performance of the developed methodology for their respective processes. The aim has been to make the development of the tools as insightful as possible in order to allow easy application and adaptation to these other industries. The level of abstraction in the definitions of tenets as binary features, entitlement calculation and system state vector is of such a high level that this should be relatively straightforward, especially given the exemplary procedure of customization for Philips Lighting.

Bibliography

[Apl01] D.W. Apley and J. Shi. A factor-analysis method for diagnosing variability in multivariate manufacturing processes. *Technometrics*, 43(1):84–95, 2001.

[Bec95] T.G. Beckwith, R.D. Marangoni, and J.H. Lienhard. *Mechanical Measurements.* Addison-Wesley Publishing Company, Massachusetts, 1995.

[Bis03] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, 2003.

[Ceg96] Member IEEE D. Ceglarek and J. Shi. Fixture failure diagnosis for autobody assembly using pattern recognition. *Transactions of the ASME, Journal of Engineering for Industry*, 118:55–66, 1996.

[Che97] P. Chen. Punch condition monitoring in sheet metal stamping under progressive stamping environments. *Internal Publication of University of Michigan, Ann Arbor*, 1997.

[Das06] Member IEEE B. Das. Fuzzy logic-based fault-type identification in unbalanced radial power distribution system. *IEEE Transactions on power delivery*, 21(1):278–285, 2006.

[Din02] Y. Ding, Member IEEE D. Ceglarek, and J. Shi. Fault diagnosis of multistage manufacturing processes by using state space approach. *Journal of Manufacturing Science and Engineering*, 124:313–322, 2002.

[Din03a] Y. Ding, P. Kim, Member IEEE D. Ceglarek, and J. Jin. Optimal sensor distribution for variation diagnosis in multistation assembly processes. *IEEE Transactions on Robotics and Automation*, 19(4):543–556, 2003.

[Din03b] Y. Ding, S. Zhou, and Y. Chen. A comparison study of process variation estimators for in-process dimensional measurements and control. *ASME Transactions, Journal of Dynamic Systems, Measurements and Control*, 2003.

[Gle97] S.S. Gleason, K.W. Tobin, and T.P. Karnowski. An integrated spatial signature analysis an automatic defect classification system. *Internal Publication of Oak Ridge National Laboratory*, 1997.

[Gro96] M.P. Groover. *Fundamentals of Modern Manufacturing.* Prentice Hall, New Jersey, 1996.

[Hay99] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice-Hall, Englewood Cliffs, New Jersey, 1999.

[Hop01] W.J. Hopp and M.L. Spearman. *Factory Physics.* Irwin McGraw-Hill, Boston, 2001.

[Hua01] Q. Huang, S. Zhou, and J. Shi. Diagnosis of multi-operational machining processes through variation propagation analysis. *Internal Publication of department of industrial and operations engineering, University of Michigan, Ann Arbor*, pages 1–10, 2001.

[Jin99] J. Jin. Feature extraction of waveform signals for stamping process monitoring and fault diagnosis. *Internal Publication of University of Michigan, Ann Arbor*, 1999.

[Jin00] J. Jin and J. Shi. Diagnostic feature extraction from stamping tonnage signals based on design of experiments. *Transactions of the ASME*, 122:360–369, 2000.

[Joh98] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis.* Prentice-Hall, New Jersey, 1998.

[Kha99] A. Khan, Member IEEE D. Ceglarek, J. Shi, J. Ni, and T.C. Woo. Sensor optimization for fault diagnosis in single fixture systems, a methodology. *ASME Journal of Manufacturing Science Engineering*, 121:109–121, 1999.

[Koh95] C.K.H. Koh, J. Shi, and W. Williams. Tonnage signature analysis using the orthogonal (haar) transforms. *NAMRI/SME Transactions*, 23:229–234, 1995.

[Law01] A.M. Law and W.D.Kelton. *Simulation modeling and analysis.* McGraw-Hill, 2001.

[Lin64] G.F. Lindsay and A.B. Bishop. Allocation of screening inspection effort - a dynamic programming approach. *Management Science*, 10:342–352, 1964.

[McC98] J.T. McClave, P.G. Benson, and T. Sinchich. *Statistics for Business and Economics.* Prentice Hall International, London, 1998.

[Men95] Fellow IEEE J.M. Mendel. Fuzzy logic systems for engineering: A tutorial. *Proceeding of the IEEE*, 83(3):345–377, 1995.

[Min05] Y. Minowa. Classification rules discovery from selected trees for thinning with the c4.5 machine learning system. *The Japanese forest society and Springer-Verlag Tokyo*, 10:221–231, 2005.

[Mon99] D.C. Montgomery and G.C. Runger. *Applied Statistics and Probability for Engineers.* John Wiley & Sons, New York, 1999.

[Ott00] E.R. Ott, E.G. Schilling, and D.V. Neubauer. *Process Quality Control.* McGraw-Hill, New York, 2000.

[Pac04] M. Pacella, Q. Semeraro, and A. Anglani. Adaptive resonance theory-based neural algorithms for manufacturing process quality control. *International Journal of Production Research*, 40(21):4581–4607, 2004.

[Rao98] C.R. Rao and J. Kleffe. *Estimation of Variance Components and Applications.* North-Holland, Amsterdam, 1998.

[Ree05] J.E.v. Reede and J. Boonekamp. Analysis of gls production data. *Technical Note*, (71082/00):1–23, 2005.

[Rob95] T. Robbins. Signature-based process control & spc trending evaluate press performance. *Metal Forming*, pages 44–50, 1995.

[Ros00] S.M. Ross. *Probability Models.* Academic Press, San Diego, 2000.

[She80] W. Shewhart. *Economic Control of Quality of Manufactured Product.* American Society for Quality Control, Milwaukee, Wisconsin, 1980.

[Tob97] K.W. Tobin, S.S. Gleason, F. Lakhani, and M.H. Bennett. Automated analysis for rapid defect sourcing and yield learning. *Future Fab International*, 4:1–15, 1997.

[Udw94] F.E. Udwadia. Methodology for optimum sensor locations for parameter identification in dynamic systems. *Journal of Mechanical Engineering*, 102:368–390, 1994.

[Whe95] D.J. Wheeler. *Advanced Topics In Statistical Process Control.* SPC Press, Knoxville, Tennessee, 1995.

[Win98] W. Winiwarter. Lazy learning algorithms for problems with many binary features and classes. *Iberamia*, pages 112–124, 1998.

[Yum81] B.J. Yum and E.D. McDowell. The optimal location of inspection effort in a class of nonserial production systems. *IIE Transactions*, 13:285–293, 1981.

[Zho00] S. Zhou, Q. Huang, and J. Shi. State space modeling for dimensional monitoring of multistage machining process using differential motion vector. *IEEE Transactions on Robotics and Automation*, 19(2):296–309, 2000.

[Zho03] S. Zhou, Y. Chen, and J. Shi. Statistical estimation and testing for variation root cause determination of multistage manufacturing processes. *Internal Publication of department of industrial engineering, University of Wisconsin*, pages 1–27, 2003.

Appendix A

Camera tool

The pulses of the camera tool are triggered via the software package *Wonderware*. Figure A.1 illustrates the connection between the control system and the camera tool.

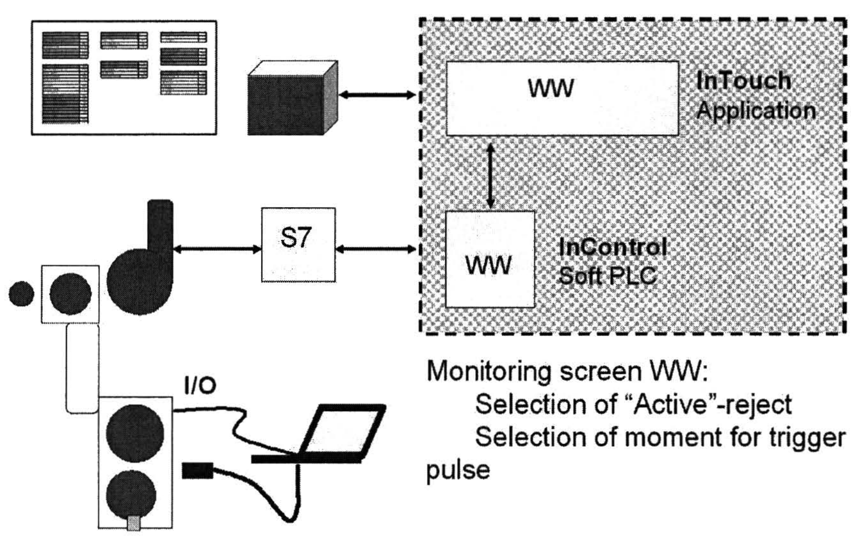

Figure A.1: Overview of connection between Wonderware application and Rejana

The pulse signals are illustrated in Figure A.2.

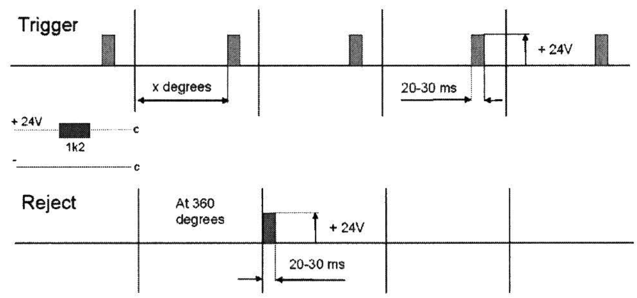

Figure A.2: Pulses that are sent from the control system to the camera

A labview application has been built. A user-interface has been developed in order to be able to adjust the diverse settings (Figure A.3).

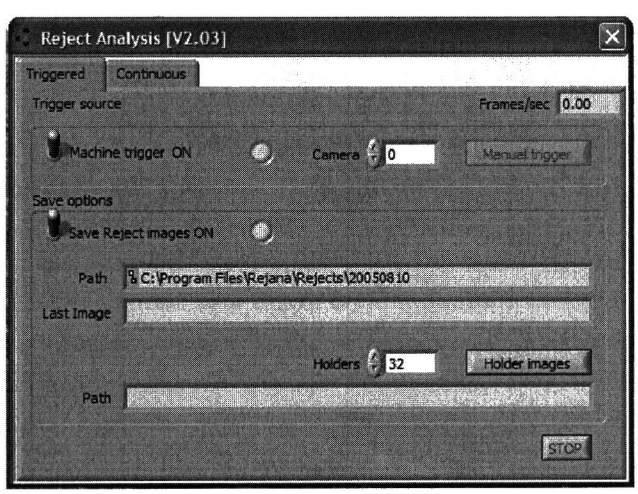

Figure A.3: Operating window in the user-interface

Appendix B

Training cases

In this appendix the state vectors $\mathbf{s}_t$ of the training cases are presented. These vectors are automatically delivered by the analysis tool in Excel.

The state vectors are categorized per root-cause class. This information is retrieved by employing the camera tool during the corresponding time periods.

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurrences:
0	1	1	0	0	0	0	1	0	0	0	3
0	1	1	0	0	1	0	1	0	0	0	1
1	1	1	0	0	0	0	1	0	0	0	1
0	1	1	0	1	0	0	1	0	0	1	1

Table B.1: State vectors during training period for root-cause: rc_1 = bent lead-in-wires. The values of all binary features and amount of occurrences of the state vector are indicated

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurrences:
0	0	1	0	0	0	0	0	0	0	0	8

Table B.2: rc_2 = misaligned coil on drum, bad coil transfer system or lead-in-wires broken

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurrences:
0	0	0	1	0	1	1	0	1	0	0	2
0	0	0	0	0	0	1	0	0	0	0	2
0	0	0	0	1	1	1	0	1	0	0	1
0	0	0	0	0	1	1	0	1	0	0	1

Table B.3: rc_3 = misaligned support wire rolling process

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurrences:
0	0	0	1	0	0	0	0	0	1	0	5
0	0	1	1	0	0	0	0	0	1	0	3
1	0	0	1	0	0	0	0	0	1	0	1

Table B.4: rc_4 = malfunctioning insertion or supply of support wire 1

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurrences:
0	0	0	0	1	0	0	0	0	0	1	5
0	0	0	0	1	1	0	0	0	0	0	1

Table B.5: rc_5 = malfunctioning insertion or supply of support wire 2

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurrences:
0	0	0	0	0	0	0	0	0	0	0	167
1	0	0	0	0	0	0	0	0	0	0	9
0	0	0	0	0	0	1	0	0	0	0	8
0	1	0	0	0	0	0	0	0	0	0	6
0	0	0	0	1	0	0	0	0	0	1	5
0	0	1	0	0	0	0	0	0	0	0	4
0	0	0	1	0	0	0	0	0	1	0	4
0	0	0	0	0	1	0	0	0	0	0	2
0	0	0	0	1	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	0	1

Table B.6: rc_6 = no root-causes present. The training cases below the line have not been included in the database of the comparison algorithm (see Section 5.5.2 for explanation)

Appendix C

Matlab m-file of algorithm

```
% DATA
% classes is a cell array of Ni class matrices (double array) C_i.
% C_i consist of Nt training cases xT_it.
% xT_it is a row vector containing Nk feature values f_itk (1 or 0).
%
% INPUT
% x is a vector to be classified.
%
% OUTPUT
% simclass is a row vector containing the indices of the class x is
% assigned to and training case with which it has highest similarity
% (it is a matrix in case of equal similarity to more than one
% training case).
% sim is the maximum similarity value.

clear all
close all
clc

%% put training cases below

classes{1}=[0 1 1 0 0 0 0 1 0 0 0; % included 3 times
            0 1 1 0 0 1 0 1 0 0 0;
            1 1 1 0 0 0 0 1 0 0 0;
            0 1 1 0 1 0 0 1 0 0 1]

classes{2}=[0 0 1 0 0 0 0 0 0 0 0] % included 8 times
```

```
classes{3}=[0 0 0 1 0 1 1 0 1 0 0; % included 2 times
            0 0 0 0 0 0 1 0 0 0 0; % included 2 times
            0 0 0 0 1 1 1 0 1 0 0;
            0 0 0 0 0 1 1 0 1 0 0]

classes{4}=[0 0 0 1 0 0 0 0 0 1 0; % included 5 times
            0 0 1 1 0 0 0 0 0 1 0; % included 3 times
            1 0 0 1 0 0 0 0 0 1 0]
classes{5}=[0 0 0 0 1 0 0 0 0 0 1; % included 5 times
            0 0 0 0 1 1 0 0 0 0 0]
classes{6}=[0 0 0 0 0 0 0 0 0 0 0; % included 167 times
            1 0 0 0 0 0 0 0 0 0 0; % included 9 times
            0 0 0 0 0 0 1 0 0 0 0; % included 8 times
            0 1 0 0 0 0 0 0 0 0 0] % included 6 times

%% determine number of features
size(classes{1});Nk=ans(2);

%% init determine x-independent values
Ni=length(classes);

%% determine x-independent values
for i=1:Ni  % loop over classes
    size(classes{i});Nt(i)=ans(1);
    pf(i,:)=sum(classes{i})./Nt(i);
          % fraction (f=1) / (all f), per feature, per class
    pf
end

    %% classify inputs until ready
    ready=0;
    while ready==0
        clc
        x=input('Enter vector to classify. ');

       %% init classification loop
       i=1;
       t=1;
       simnow=0;
       class=classes{i};
       for k=1:Nk % loop over features
           if x(k)==1
               if class(t,k)==1
```

```
                simnow=simnow + pf(i,k);
            elseif class(t,k)==0
                simnow=simnow - (1 - pf(i,k));
            end
        elseif x(k)==0
            if class(t,k)==1
                simnow=simnow - pf(i,k);
            end
        end
    end
    sim=simnow;
    simclass=[i,t];

    %% classification loop
    for i=1:Ni % loop over classes
        class=classes{i};
        for t=1:Nt(i) % loop over training cases
            simnow=0;
            for k=1:Nk % loop over features
                if x(k)==1
                    if class(t,k)==1
                        simnow=simnow + pf(i,k);
                    elseif class(t,k)==0
                        simnow=simnow - (1 - pf(i,k));
                    end
                elseif x(k)==0
                    if class(t,k)==1
                        simnow=simnow - pf(i,k);
                    end
                end
            end
            if simnow>sim(1)
                sim=simnow;
                simclass=[i,t];
            elseif simnow==sim(1) & length(find(simclass==i))==0
                sim=simnow;
                simclass=[simclass;i,t];
            end
        end
    end
    sim
    simclass
    ready=input('Finished classifying? [0=no, 1=yes] ');
end
```

Appendix D

Test cases

In this Appendix the test case system state vectors $\mathbf{s}_{te_t}$ are included.

The vectors are categorized by root-cause class of the root-cause that was actually present. It is indicated in the tables to which root-cause class rc_i the state vector was assigned by the comparison algorithm.

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurr.:	class. to:
0	1	1	0	0	0	0	1	0	0	0	5	rc_1
0	1	1	0	0	1	0	1	0	0	0	2	rc_1
0	1	1	0	1	0	0	1	0	0	1	1	rc_1
0	0	1	0	0	0	0	0	0	0	0	1	rc_2

Table D.1: Test case state vectors $\mathbf{s}_{te_t}$ for actual root-cause: rc_1 = bent lead-in-wires. The present binary features are indicated, as well as the amount of occurrences of the test vector. In the last column the class to which the test case was actually classified is presented

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurr.:	class. to:
0	0	1	0	0	0	0	0	0	0	0	10	rc_2
0	0	1	1	0	0	0	0	0	1	0	1	rc_4
0	0	1	0	0	1	0	0	0	0	0	1	rc_2

Table D.2: Actual root-cause: rc_2 = misaligned coil on drum, bad coil transfer system or lead-in-wires broken

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurr.:	class. to:
0	0	0	0	1	1	1	0	1	0	0	3	rc_3
0	0	0	0	0	1	1	0	1	0	0	3	rc_3
0	0	0	1	0	1	1	0	1	0	0	1	rc_3
0	0	0	0	0	0	1	0	0	0	0	1	rc_3

Table D.3: Actual root-cause: rc_3 = misaligned support wire rolling process

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurr.:	class. to:
0	0	1	1	0	0	0	0	0	1	0	3	rc_4
0	0	0	1	0	0	0	0	0	1	0	1	rc_4
0	1	1	1	0	0	0	1	0	1	0	1	rc_1

Table D.4: Actual root-cause: rc_4 = malfunctioning insertion or supply of support wire 1

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurr.:	class. to:
0	0	0	0	1	0	0	0	0	0	1	2	rc_5
0	0	0	0	1	1	0	0	0	0	0	1	rc_5
0	0	0	0	1	0	1	0	0	0	1	1	rc_5

Table D.5: Actual root-cause: rc_5 = malfunctioning insertion or supply of support wire 2

bf_1	bf_2	bf_3	bf_4	bf_5	bf_6	bf_7	bf_8	bf_9	bf_{10}	bf_{11}	occurr.:	class. to:
0	0	0	0	0	0	0	0	0	0	0	155	rc_6
0	1	0	0	0	0	0	0	0	0	0	10	rc_6
1	0	0	0	0	0	0	0	0	0	0	7	rc_6
0	0	0	0	0	0	1	0	0	0	0	4	rc_6
0	0	0	1	0	0	0	0	0	1	0	4	rc_4
0	0	1	0	0	0	0	0	0	0	0	3	rc_2
0	0	0	0	1	1	0	0	0	0	0	2	rc_5
0	0	0	0	1	0	0	0	0	0	1	2	rc_5
1	1	0	0	1	0	0	0	0	0	1	1	rc_5
0	0	0	0	0	1	0	0	0	0	0	1	rc_5
1	1	1	0	0	0	0	0	0	0	0	1	rc_1

Table D.6: Actual root-cause: rc_6 = no root-causes present

Appendix E

Matlab m-file of alternative algorithms

The two alternative algorithms are presented here in one m-file. These algorithms calculate the selectivity, a certain measure of predictive power, of each feature. If, for instance, for class i a certain feature k is included in all training cases the selectivity of this feature for this class is said to be 1. On the other hand, if this feature would be excluded as often as included (both 50% of the cases) the selectivity is set at 0.

As such, these algorithms assess the importance of features for a certain class and take this into account while assessing the similarity of a new case to training cases.

This selectivity can either be averaged over all classes (run with average==1), or be determined per class (run with average==0).

```
% DATA
% classes is a cell array of Ni class matrices (double array) C_i.
% C_i consist of Nt training cases xT_it.
% xT_it is a row vector containing Nk feature values f_itk (1 or 0).
%
% INPUT
% x is a vector to be classified.
%
% OUTPUT
% simclass is a row vector containing the indices of the class x is
% assigned to and training case with which it has highest similarity
% (it is a matrix in case of equal similarity to more than one
% training case).
% sim is the maximum similarity value.

clear all
```

```
close all
clc

%% put training cases below

classes{1}=[0 1 1 0 0 0 0 1 0 0 0; % included 3 times
            0 1 1 0 0 1 0 1 0 0 0;
            1 1 1 0 0 0 0 1 0 0 0;
            0 1 1 0 1 0 0 1 0 0 1]

classes{2}=[0 0 1 0 0 0 0 0 0 0 0] % included 8 times

classes{3}=[0 0 0 1 0 1 1 0 1 0 0; % included 2 times
            0 0 0 0 0 0 1 0 0 0 0; % included 2 times
            0 0 0 0 1 1 1 0 1 0 0;
            0 0 0 0 0 1 1 0 1 0 0]

classes{4}=[0 0 0 1 0 0 0 0 0 1 0; % included 5 times
            0 0 1 1 0 0 0 0 0 1 0; % included 3 times
            1 0 0 1 0 0 0 0 0 1 0]
classes{5}=[0 0 0 0 1 0 0 0 0 0 1; % included 5 times
            0 0 0 0 1 1 0 0 0 0 0]
classes{6}=[0 0 0 0 0 0 0 0 0 0 0; % included 167 times
            1 0 0 0 0 0 0 0 0 0 0; % included 9 times
            0 0 0 0 0 0 1 0 0 0 0; % included 8 times
            0 1 0 0 0 0 0 0 0 0 0] % included 6 times


%% determine number of features
size(classes{1});Nk=ans(2);

%% init determine x-independent values
Ni=length(classes);

%% determine x-independent values
for i=1:Ni  % loop over classes
    size(classes{i});Nt(i)=ans(1);
    pf(i,:)=sum(classes{i})./Nt(i);
      % fraction (f=1) / (all f), per feature, per class
    pf
end

S=1-4.*pf.*(1-pf); % selectivity = 1 for all f_itk = 1 or 0
%               selectivity = 0 for 50% of f_itk = 1, or 50% of f_itk = 0
```

```
average=input('Do you want to average selectivity?
-- [1=yes, average over classes / 0=no, determine selectivity per class] ');

if average==0
    w=S;
elseif average==1
    w=mean(S); % average S over classes, per feature
end

if average==1

    %% classify inputs until ready
    ready=0;
    while ready==0
        clc
        x=input('Enter vector to classify. ');

       %% init classification loop
       i=1;
       t=1;
       simnow=0;
       class=classes{i};
       for k=1:Nk % loop over features
           if x(k)==1
               if class(t,k)==1
                   simnow=simnow + pf(i,k)*w(k);
               elseif class(t,k)==0
                   simnow=simnow - (1 - pf(i,k))*w(k);
               end
           elseif x(k)==0
               if class(t,k)==1
                   simnow=simnow - pf(i,k)*w(k);
               end
           end
        end
        sim=simnow;
        simclass=[i,t];

        %% classification loop
        for i=1:Ni % loop over classes
            class=classes{i};
            for t=1:Nt(i) % loop over training cases
                simnow=0;
                for k=1:Nk % loop over features
```

```
                    if x(k)==1
                        if class(t,k)==1
                            simnow=simnow + pf(i,k)*w(k);
                        elseif class(t,k)==0
                            simnow=simnow - (1 - pf(i,k))*w(k);
                        end
                    elseif x(k)==0
                        if class(t,k)==1
                            simnow=simnow - pf(i,k)*w(k);
                        end
                    end
                end
                if simnow>sim(1)
                    sim=simnow;
                    simclass=[i,t];
                elseif simnow==sim(1) & length(find(simclass==i))==0
                    sim=simnow;
                    simclass=[simclass;i,t];
                end
            end
        end
        sim
        simclass
        ready=input('Finished classifying? [0=no, 1=yes] ');
    end

elseif average==0

    %% classify inputs until ready
    ready=0;
    while ready==0
        clc
        x=input('Enter vector to classify. ');

       %% init classification loop
       i=1;
       t=1;
       simnow=0;
       class=classes{i};
       for k=1:Nk % loop over features
           if x(k)==1
               if class(t,k)==1
                   simnow=simnow + pf(i,k)*w(i,k);
               elseif class(t,k)==0
```

```
                simnow=simnow - (1 - pf(i,k))*w(i,k);
            end
        elseif x(k)==0
            if class(t,k)==1
                simnow=simnow - pf(i,k)*w(i,k);
            end
        end
    end
    sim=simnow;
    simclass=[i,t];

    %% classification loop
    for i=1:Ni % loop over classes
        class=classes{i};
        for t=1:Nt(i) % loop over training cases
            simnow=0;
            for k=1:Nk % loop over features
                if x(k)==1
                    if class(t,k)==1
                        simnow=simnow + pf(i,k)*w(i,k);
                    elseif class(t,k)==0
                        simnow=simnow - (1 - pf(i,k))*w(i,k);
                    end
                elseif x(k)==0
                    if class(t,k)==1
                        simnow=simnow - pf(i,k)*w(i,k);
                    end
                end
            end
            if simnow>sim(1)
                sim=simnow;
                simclass=[i,t];
            elseif simnow==sim(1) & length(find(simclass==i))==0
                sim=simnow;
                simclass=[simclass;i,t];
            end
        end
    end
    sim
    simclass
    ready=input('Finished classifying? [0=no, 1=yes] ');
  end
end
```