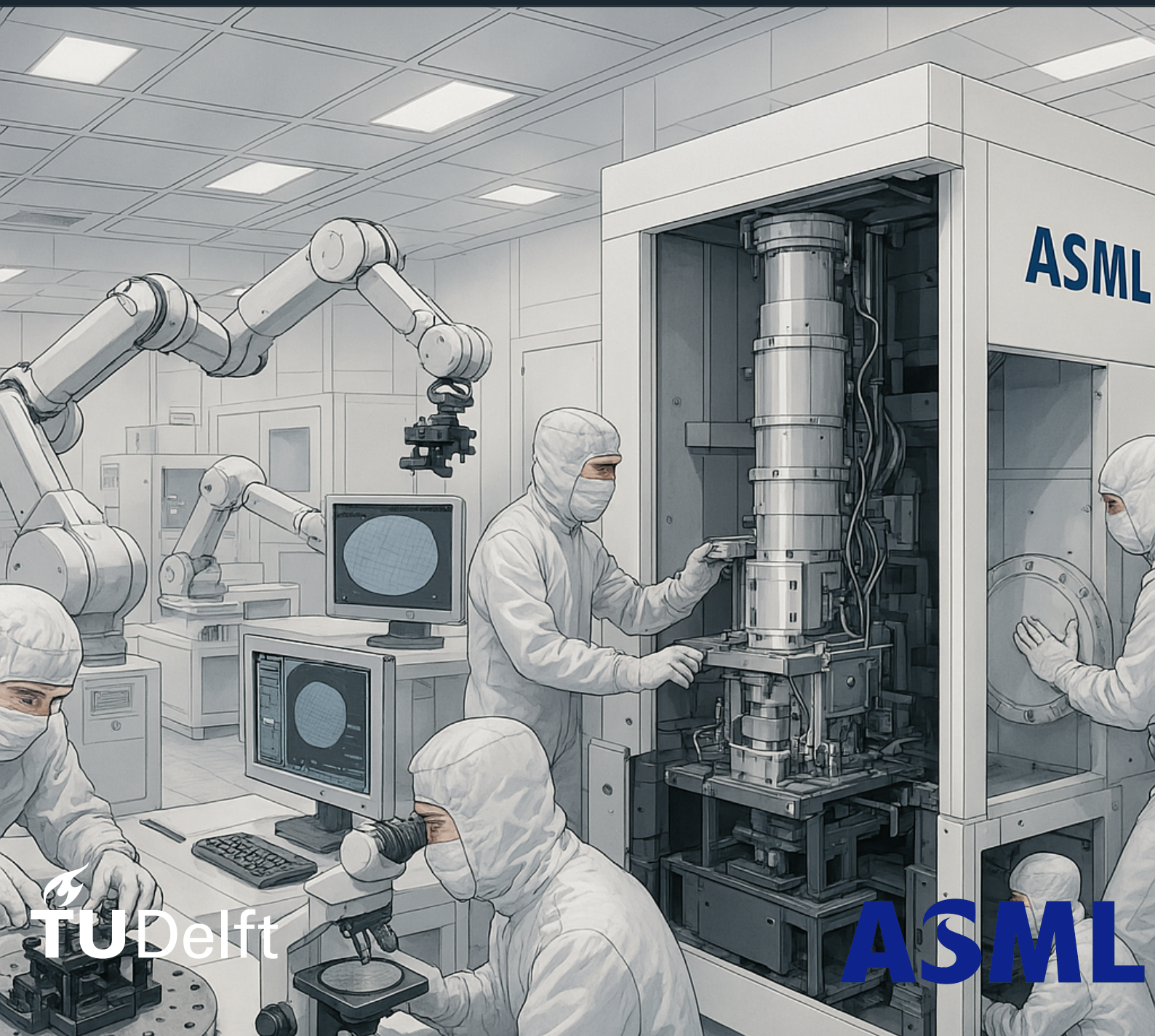
Online change detection using sensor groups for high-dimensional short-run manufacturing processes

An ASML case study

T.H.J. Beene



TU Delft

ASML

Online change detection using sensor groups for high-dimensional short-run manufacturing processes

An ASML case study

by

T.H.J. Beene

to obtain the degree of Master of Science at the Delft University of Technology, to be defended publicly on Thursday, July 3rd, 2025.

Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	September, 2024 – June, 2025
Student number:	4859316
Thesis committee:	Dr. J. Sun (chair) Dr. I.I. de Pater (supervisor) Dr. A.A. Simkooei (external member) MSc P. Scheffers (ASML supervisor)

Cover Image: Created by Microsoft's AI image generator.

Acknowledgements

This master thesis marks the end of my time as a student at Delft University of Technology. Looking back at this experience, I can say that it has been an interesting journey where I learned a lot. My enthusiasm for research has grown, and I have met many inspiring people along the way. The thesis project was carried out as part of a student internship at ASML Delft. I would like to express my sincere thanks to Irina Rod, Paul Scheffers, and Hans van Gurp for initiating the project and for their guidance and supervision throughout. I am also grateful to the entire Q-branch team at ASML Delft for their support and collaboration during the course of the project. From TU Delft, I would like to thank Ingeborg de Pater for her dedicated supervision and the many insightful conversations we shared. Her support has been important in shaping this work and furthering my interest in a career in academia. I also extend my thanks to Joris Bierkens for his helpful discussion and contribution to the mathematical aspects of the research. Finally, I am deeply thankful to my family, my girlfriend, and my friends for their support and encouragement throughout the Masters program.

Online change detection using sensor groups for high-dimensional, short-run manufacturing processes

Thijs BEENE

Abstract

Detecting change in sensor measurements is essential for maintaining product quality and ensuring efficiency in manufacturing processes. Traditionally, statistical methods such as control charts are used to detect changes by comparing new sensor measurements with historical data. However, in high-dimensional, short-run (HDSR) settings, where there are many sensors and only limited or no historical observations, change detection becomes challenging and sometimes even impossible. HDSR processes are mostly present in specialized industries where errors can be costly, such as: semiconductors, aerospace or shipping. Previous research highlights several control charts to address HDSR processes and also demonstrated that grouping sensors can improve change detection. Finding groups of sensors was done by incorporating expert knowledge or by combining similar sensor data to increase sample size. In this research, a novel procedure for finding groups of sensors is proposed, by using an algorithm that automatically groups sensors based on the maximization of the probability of detection. The procedure and three state-of-the-art alternatives are applied to a case study involving a semiconductor manufacturing process of a new electron optical module. The results reveal that the proposed procedure finds groups of sensors that reflect sensor covariance and process knowledge. Furthermore, it is shown that the probability of detecting persistent mean shifts is improved compared to the three alternative control charts. Specifically, the proposed procedure had faster detection of shifts and also a higher POD for small magnitude shifts. Areas for future research could be the extension of the proposed procedure to a Bayesian framework.

KEYWORDS AND PHRASES: Statistical process monitoring, Self-starting control charts, Multivariate statistics, Bayesian statistics, Change detection.

NOMENCLATURE

Symbol	Description
p	Number of sensors in a group
ARL_{IC}	In control average run length
POD_r	Probability of detection after r observations
$\hat{\Sigma}$	Estimated covariance matrix
Σ	Known covariance matrix
λ	Smoothing constant for SSMEWMA
n	Number of observations
m	Number of charts
$\bar{h}$	Limit for HC chart
h	Limit for SSMEWMA chart
k	Limit for Q-chart
δ	Shift magnitude
N	Number of runs
τ	Start of OOC
α	False alarm rate

1. INTRODUCTION

Statistical process monitoring (SPM) is a method that makes use of statistical tools and techniques for the management and improvement of processes [1]. SPM considers that all processes display a combination of random and non-random behavior. When a process only has random variation, it is behaving normal and in-control (IC) [2]. A process is out-of-control (OOC) when external factors cause non-random variation [2]. In SPM, control charts are used to detect non-random variation or signal process deterioration and label them as an OOC event [3]. Usually control charts are designed based on parameter estimates of the mean and variance, which adds a random element to the control chart and can affect control chart performance [4].

'Traditional' SPM control charts are normally implemented for applications when it is possible to obtain at least 20 to 50 observations of process quality parameters in a short time for parameter estimation, otherwise self-starting charts based on an unknown parameter estimate are used [5][6]. Quality parameters are considered in a broad context, and can for instance be part diameter, temperature or pressure measurement. Sensors measure these quality parameters Self-starting control charts transform the stream of unknown-parameter data into a corresponding stream of standard normal process readings, removing the problem of unkown parameters [7].

In practice, process data is often not univariate. Real-time process data is likely to be correlated. Multivariate

control charts should be used for maximum effectiveness when multiple correlated time-series are monitored [7]. For producing high-quality products, continuous monitoring of many critical-to-quality (CTQ) parameters is vital [8]. Several multivariate control charts exist, such as the Hotelling T^2 and the multivariate exponentially weighted moving average (MEWMA).

For high-dimensional, short-run (HDSR) processes, which are often prevalent in industries such as aerospace, shipping and semiconductors, it is difficult to implement SPM. An example of such a process could be the manufacturing of a new type of aircraft. These processes are often characterized by complex, expensive and customized parts, for which SPM is usually considered to not even be applicable [9]. Mainly because the sample size is small and many parameters could be relevant to product quality.

To overcome these issues and enable the application of SPM to low volume processes, groups of data can be defined [10]. Not many standards regarding the grouping of process data exist and the ones that do exist are vague in describing the application of a grouping strategy. For instance, the ISO 7870-8 [6] and BS 5702-3 [11] standards suggest that the grouping of data can be beneficial for short run processes. Yet both standards do not provide exact methods of doing so.

Grouping of similar process data was shown to allow for the assessment of the significance of process influences using a cost-efficient procedure based on expert knowledge [9]. However, this procedure alone did not reveal a significant factor that was identified by the analysis of historical data and its informative value is low [9]. Furthermore, this method requires a lot of knowledge, work force and data preprocessing [10]. While for some processes, expert knowledge might not even be present, or be flawed, leading to a sub optimal clustering of sensors. Complementing expert knowledge with an algorithm that extracts significant influences could be beneficial [9].

Work by Greipel et al. [10] builds on the idea of increasing sample size by grouping data (ISO 7870-8 [6]) without expert knowledge. They compared four different clustering algorithms for application to small sample sizes. However, their approach viewed clustering only as a means to increase sample size, by for instance grouping measurements of diameters (features) with similar statistical metrics (means and variances) and monitoring this data in the same univariate moving range or individuals control charts. This approach still requires historical measurement data of features and feature categories and is incapable of considering relations between sensors that measure different quantities.

The objective of this research is designing a sensor grouping algorithm to be used for online monitoring of HDSR processes. The procedure is applied to a case study of the high-tech workcenter for electron optical module assembly, where currently a manufacturing process for a new electron optical module is being designed and implemented. An SPM system for this new manufacturing process is to be designed as well, which has the potential to significantly reduce costs, increase module quality and contribute to sustainable resource management.

The process is monitored by many potentially correlated sensors and is considered to produce modules at a low throughput. Historical data that can provide exact parameter estimates is not available due to a change in specifications. Emphasis is placed on fast detection of OOC events, since preventing expensive and time consuming errors outweighs the cost incurred by investigating false positives.

Specifically, we attempt to improve performance by combining the SSMEWMA chart [7] with a sensor grouping algorithm that attempts to find the optimal partition of process data. This algorithm can be used for real-time group determination. In total, we apply four control charts to the historical data of the previous high-tech workcenter for electron optical module assembly process. The following charts will be used: Q-chart [12], HC chart [8], SSMEWMA with random cluster assignments (R-SSMEWMA) [7] and the SSMEWMA with our partitioning algorithm (P-SSMEWMA). A Monte Carlo simulation is performed to determine the performance of the charts. This research only considers the performance with respect to a persistent mean shift in a single sensor.

This article is structured as follows: first we give background about the case study and describe the data used. Next in the theory, we explain general metrics and give definitions used throughout the article. This is followed up by the mathematical descriptions of the control charts and the proposed procedure for partitioning the sensors. Next the methodology is divided into two parts, part 1 explains the steps taken to verify the new partitioning algorithm and the implementation of the control charts. Part 2 highlights the application of the methods to a specific case study. In the results and discussion section, we first show results from the verification of the partitioning algorithm. Next, the main results of the control chart applied to the case study are shown and we finish with the results from a sensitivity analysis. Finally, the conclusions and recommendations for future work are given.

1.1 Case study: High-tech workcenter for electron optical module assembly

A new generation wafer scanning electron microscope (SEM) is planned that follows up on the HMI eScan 1100, see Figure 1. Part of this new scanner is the electron optical module 2. The assembly line for this module is currently being designed and build in the cleanroom of the high-tech workcenter. During the manufacturing process, sensor readings provide insights about process state and product quality. Monitoring these sensors and signaling on OOC events is therefore important for ensuring proper machine quality. An example of an OOC event could be a persistent shift in the mean of the diameter of a module part. This could for

instance be caused by a calibration error. Another example could be a shift in part cleanliness caused by the introduction of a new manufacturing procedure. Detection speed is also relevant for module quality. If detection speed is low, it could take multiple extra observations before an OOC event is detected. During this time, several low-quality modules could be produced.

Figure 1: HMI eScan 1100 multibeam inspection tool [13].

1.1.1 Data description

Data from the optical module of the eScan 1100 is available and resembles optical module 2 data. In total process data of 32 electro-optical modules 1 was recorded in the data set. This means that a total of 32 observations are available. Multiple sensor measurements of quality parameters were registered, however not all were taken into account for these simulations. First of all, some quality data was simply reported as an OK/NOK, which is not feasible for numerical simulations. Furthermore, some parameters were constant across all observations, these were also not taken into account. After data preprocessing a list with a total of 96 quality parameters (p) and 32 observations (n) remained. Dimensionality reduction methods were not applied to the data set, as this would leave out sensors that could potentially give an OOC signal.

The process data contains correlated sensors due to relations and/or dependencies between different sensors. For instance, groups of sensors are related to steps in the process. When a process shift occurs during one of these steps, it is likely that multiple sensors related to the step register the shift. Furthermore, shifts could also be caused by common influences such as temperature or particle count, a shift caused by a common influence could be reflected in multiple sensors.

2. THEORY

In this section we explain the relevant theory. First, we discuss metrics used to measure control chart performance and compare different control charts. Next, we give the derivations of the statistics of three control charts. Lastly, we propose a new algorithm combining the SSMEWMA and Q-chart based on an optimized data partitioning, referred to as the P-SSMEWMA.

2.1 Metrics and definitions

In order to compare the four control charts, appropriate performance metrics should be defined. First we define our sequence of data. Let $X_{i,j}$ for $i = 1, 2, ..., n$ and $j = 1, 2, ..., p$ be a multivariate sequence of observations from a process with p sensors and covariance Σ and mean vector μ. A persistent mean shift occurring at observation $n = \tau$ is defined as:

$$x_i = \begin{cases} \mathcal{N}(\mu, \Sigma) & \text{for } n < \tau \\ \mathcal{N}(\mu + \delta, \Sigma) & \text{for } n \geq \tau \end{cases} \tag{2.1}$$

Here, the process before τ is IC and the process after is OOC. See Figure 2 for an IC and OOC process. The goal of control charts is to detect this process shift from IC to OOC. A control chart gives a signal when the monitored observations or statistic exceed either the upper or lower control limits (UCL and LCL). Multiple other process shifts can be identified, such as outliers, trends, variance shifts and cyclic patterns. However, in this research we only consider a persistent mean shift.

The average run length (ARL) is often used to obtain an overview of the control charts performance. This is the expected value of the run length before signaling an OOC across multiple runs. A distinction can be made between the in-control ARL (ARL_{IC}) and out-of-control ARL (ARL_{OOC}). The ARL depends on the shift magnitude (δ) and the observation at which the shift occurred (τ). The ARL_{IC} and ARL_{OOC} are related to the type 1 and type 2 errors in hypothesis testing respectively. See Figure 2 for an example of an in-control and out-of-control process and the related metrics.

However, as discussed by Quessenberry and Laurijsse [14][15], the ARL is often not sufficient for determining control chart's responsiveness to OOC shifts. The main issue lies in the detection of small shifts. When the ARL is used, enough observations after the shift must be generated until the chart signals an OOC. This increases computational times significantly and also increases the likelihood of a false signal. To mitigate these issues, the probability of detection (POD) can be used, see Figure 3.

For calculating the POD, we need the expected delay (E) and run length (RL). Where RL is the number of observations until some event occurs. E is the expected number of observations after a shift at observation τ starting from a value of 1, E can be seen as the out-of-control run length. We adopt the same definition for E as used by Li et al [17]:

$$E_\delta = RL - \tau + 1 | RL \geq \tau \tag{2.2}$$

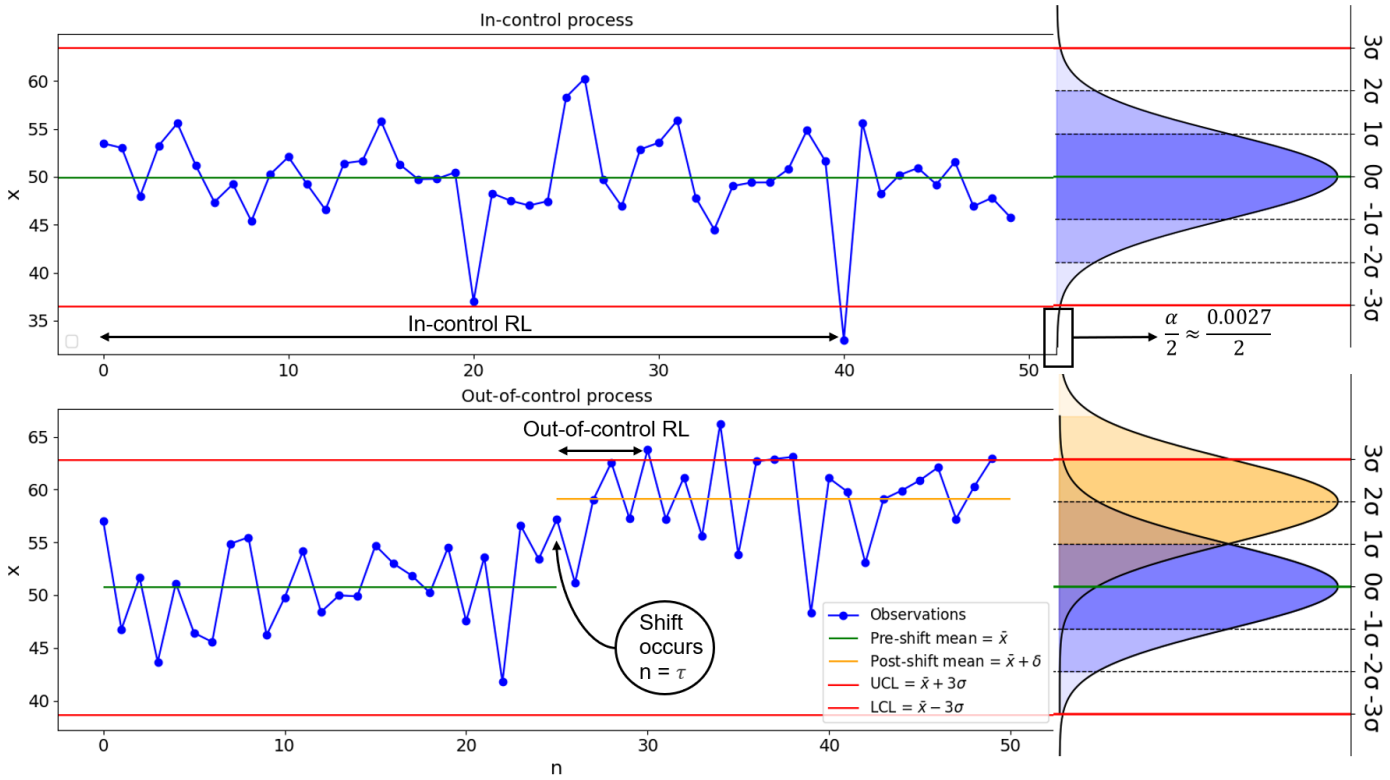


Figure 2: In-control and out-of-control charts with metrics indicated. Fixed control limits are used. A persistent mean shift is shown.

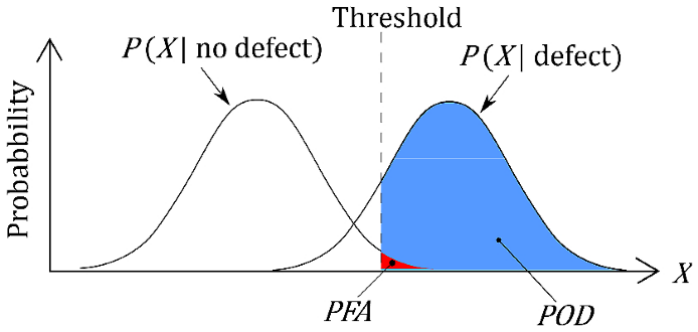


Figure 3: Visual illustration of probability of detection (POD), probability of false alarm (PFA) (also written as α) and the probability of an observation X given that the process is IC (P(X|no defect)) and the probability of an observation X given that the process is OOC (P(X|defect)). The threshold indicates a control limit for our application. Figure taken from [16].

Where 1 is added to account for $RL = \tau$, i.e. an immediate detection of a shift. This is conventional, since we would otherwise be able to detect a shift after 0 observations. We can view the ARL_{OOC} as the expected value of E over many runs: $ARL_{OOC}(\delta) = E(E_\delta) = \frac{E_\delta}{N}$, with N the total number of runs performed. When performing N runs, we count how many times E_δ is lower than a given number of observations r, for example, 2,5 and 10. We define the POD of a shift δ at observation τ before r extra observations for a statistic x (could be the observations) with limit L as:

$$POD_r = P(|x| > L|\delta, \tau, n \leq r) = \frac{1}{N}\sum_{i=1}^{N} g(i)_{(0<E_{\delta,i}\leq r)} \tag{2.3}$$

Where i denotes a single run and:

$$g(i)_{(0<E_{\delta,i}\leq r)} = \begin{cases} 1 & \text{if } 0 < E_{\delta,i} \leq r \\ 0 & \text{else} \end{cases} \quad \forall i \in N$$

The POD_r is a proportion with following corresponding standard error (SE):

$$SE = \sqrt{\frac{p_r(p_r - 1)}{N}} \tag{2.4}$$

where p_r represents the subset for a given r of the count of values in a set N ($p_r = \sum_{i=1}^{N} g(i)_{(0<E_{\delta,i}\leq r)}$). Using the SE, the 95% confidence interval (CI) can be defined:

$$CI_{95} = \bar{X} \pm SE \tag{2.5}$$

Equation 2.5 is used to define the errors in the values found for the POD_r.

2.2 Q-chart

Self-starting control charts were first proposed by Hawkins [18]. The univariate Q-chart, was later developed and named by Quessenberry [12]. The main idea of the Q-chart is to standardize the newest measurement at observation x_n with the estimated mean and sample variance $\bar{X}_{n-1}$ and S^2_{n-1} of the assumed to be IC previous observations. This is done with the t-statistic. The t-statistic gives the difference between a samples estimated mean and measured observation, relative to the standard error. Note that these are calculated for each sensor separately. The standardized T_n statistic for the Q-chart is defined as:

$$T_n = \frac{x_n - \bar{X}_{n-1}}{S_{n-1}} \tag{2.6}$$

It is evident from equation 2.6 that the Q-chart can only start monitoring when $n \geq 3$, since at least 2 previous observations are required to estimate S_{n-1}. Quessenberry [12] shows that $\sqrt{\frac{n-1}{n}}T_n$ follows a t-distribution, specifically t_{n-2}. The uniform distribution $U_n \in$ N(0,1) can be obtained by taking the cumulative distribution function (cdf)(F_{n-2}) of $\sqrt{\frac{n-1}{n}}T_n$:

$$U_n = F_{n-2}\left(\sqrt{\frac{n-1}{n}}T_n\right) \tag{2.7}$$

Taking the inverse cumulative distribution function (CDF) (ϕ^{-1}) of the normal distribution $N(0,1)$ yields a standard normal random variable Z_n:

$$Z_n = \phi^{-1}(F_{n-2}(\sqrt{\frac{n-1}{n}}T_n)) \tag{2.8}$$

The Z_n statistic is used in the Q-chart for signaling on OOC events, a signal is given if $Z_n >$ UCL or $Z_n <$ LCL. However, the control limits have to be specified in terms of standard deviations by the user of the chart and the Z_n statistic is not intuitive to analyze. For this reason, the UCL and LCL for Z_n are often transformed back to provide updating limits on the original measurement X_n [15]. The resulting upper and lower control limits can be calculated as follows:

$$u_n = \bar{X} + F^{-1}_{n-1}(\Phi(UCL))\sqrt{\frac{n+1}{n}}S_{n-1} \tag{2.9}$$

and the lower control limit (LCL):

$$l_n = \bar{X} - F^{-1}_{n-1}(\Phi(LCL))\sqrt{\frac{n+1}{n}}S_{n-1} \tag{2.10}$$

The full derivation is given by Laurijsse [15] and starts with the relation $Z_n > UCL$. Where UCL and LCL are the control limits on Z_n. Since Z_n is independent and indentically distributed (i.i.d), UCL and LCL are given in number of standard deviations from the mean. The values for UCL and LCL are often replaced by k, which is similar to the p value in the t-test. This k value can be found with the desired false positive rate or ARL_{IC} of the chart. By taking the inverse cdf of the normal distribution of the false positive rate for both UCL and LCL, we obtain k. This holds because all Z_n statistics are normally distributed and independent when the process is in-control. The probability (α) of a signal being given when the process is IC can be found with:

$$\alpha = \Phi(LCL) + 1 - \Phi(UCL) = 2(1 - \Phi(k)) \tag{2.11}$$

Where LCL = -k and UCL = k and α is related to the ARL_{IC} as $ARL_{IC} = \frac{1}{\alpha}$.

As the number of observations increases, the Q-chart control limits converge to the standard Shewhart limits. For this reason, when 30 observations are reached, it is recommended by Laurijsse [15] to switch to the Shewhart $\bar{X}$-chart. An example of the Q-chart for a run with 30 observations and no shift is shown in Figure 4.

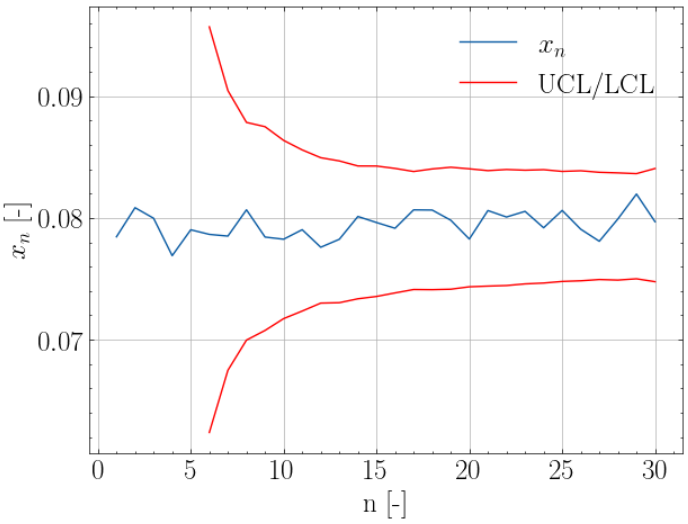


Figure 4: Q control chart [15] example for a run length of 30 observations and no shift.

2.3 HC-chart

The multivariate change point detection chart (HC) was proposed by Li et al [17]. Chen and Qin [8] defined a statistic for testing the difference between the mean vectors of the pre-shift and post-shift data with n total observations at time k:

$$W_{n,k} = \frac{\sum_{i,j=1;i\neq j}^{k} X_i^{'} X_j}{k(k-1)} + \frac{\sum_{i,j=k+1;i\neq j}^{n} X_i^{'} X_j}{(n-k)(n-k-1)} - 2\frac{\sum_{i=1}^{k}\sum_{j=k+1}^{n} X_i^{'} X_j}{k(n-k)} \tag{2.12}$$

$W_{n,k}$ can be seen as an extension of the standard likelihood ratio test using the Hotelling T^2 statistic to smaller sample sizes, as T^2 is not defined when n < p. A larger value for $W_{n,k}$ indicates that the time series until k is more different in mean from the time series after k. The main advantage of using the $W_{n,k}$ statistic is that we can monitor processes with many sensors right away. In this research, we use the variance estimate (σ_W^{2*}) developed by Chen and Qin [8]:

$$\sigma_W^{2*} = \frac{2tr(\Sigma_0^2)}{k(k-1)} + \frac{2tr(\Sigma_1^2)}{(n-k)(n-k-1)} + \frac{4tr(\Sigma_0\Sigma_1)}{k(n-k)} s_3^* \tag{2.13}$$

σ_W^{2*} is much less computationally intensive compared to the estimator S_W^{2*} [17]. However, σ_W^{2*} is not transformation invariant. The test statistic Z for finding the change point can be calculated by determining the maximum value over all possible splits:

$$Z_{max,n} = \max_{2\leq k\leq n-1} \left(\frac{W_{n,k}}{\sqrt{\Sigma_W^{2*}}} \right) \tag{2.14}$$

$\frac{W_{n,k}}{\sqrt{\Sigma_W^{2*}}}$ can be seen as a sort of signal-to-noise ratio. It is unknown when the true change-point k occurred, therefore by calculating $Z_{n,k}$ for all possible splits, we assume that every split could have been a possible change-point. Monitoring the maximum value of $Z_{n,k}$ is equivalent to monitoring the difference between the means of the data for the most likely change-point k.

Here we slightly altered the implementation from the paper by Li et al [17]. The maximum over all possible splits ranging from $2 \leq k \leq n-1$ was calculated. This was done to avoid a division by 0 error when $k = 1$. The $Z_{max,n}$ statistic signals if the control limit $h_{n,p}$ is exceeded. This alteration was confirmed to be correct by the authors of the original paper. The control limit is an asymptotic function of both p and n. It is calculated numerically by generating several $Z_{max,n}$ values for a specific combination of n and p and determining the percentile resulting in a desired false positive rate. The implemented HC control chart is shown in Figure 5 for N = 30 and no shift.

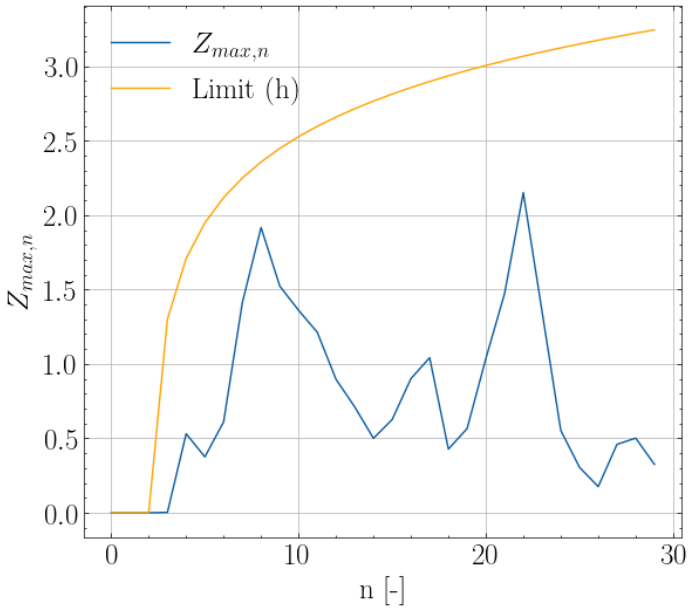


Figure 5: HC control chart [17] example for a run length of 30 observations and no shift.

2.4 SSMEWMA-chart

The self-starting multivariate exponentially weighted moving average (SSMEWMA) chart is suitable for monitoring many correlated sensors. This indicates that for a process with many sensors we could define groups of sensors monitored by SSMEWMA charts, where each group is monitored by a single SSMEWMA chart. The general multivariate method was proposed by Hawkins and Maboudou-Tchoa and can be used as a front-end for any multivariate chart [7]. We follow the implementation of their method with the MEWMA chart, as the implementation of Hotelling T^2 was found to be ineffective [7].

The transformation from unknown parameters to standard normal is similar to the univariate case used for the Q-chart. The first step is again standardizing the data. Multivariate standardization involves the transformation of X_n to a standard normal vector $Z_n = A(X_n - \mu)$. Where the matrix A satisfies $A\Sigma A' = I$. A can be found by decomposing Σ using the triangular Cholesky inverse root. By doing so, the components of Z_n are the regression residuals normalized by variance.

However, Σ and μ are generally not know for self-starting charts. Therefore, the recursive residuals are used to obtain estimates. The recursive residuals are defined as $r_{i,j}$ for $i \in [1, ..., n]$ and $j \in [1, ..., p]$, with p the number of sensors in the regression model and n the current number of observations. The $r_{i,j}$ are determined by the regression of each $x_{i,j}$ on $x_1, ..., x_{n-1}$ for $j \in p$. The following multiple regression model is solved for β_j:

$$y_{i,j} = X_{i-1,j-1}\beta_j + \epsilon_{i,j} \tag{2.15}$$

The value for β_j that minimizes $\epsilon_{i,j}$ is the maximum likelihood estimator (MLE) for the model, referred to as $\hat{\beta}_j$. The remaining value for $\epsilon_{i,j}$ evaluated at the MLE, is the residual or unexplained variance in the model. Once $\hat{\beta}_j$ has been determined, we can predict the next $\hat{x}_{i,j}$, using:

$$\hat{x}_{i,j} = x_{i,j-1}\hat{\beta}_{i-1,j} \tag{2.16}$$

see Figure 6 for a visual overview of the multiple regression procedure. Ones are prepend to the X matrix to handle the intercept. The nth recursive residual is defined as:

$$r_{i,j} = \frac{x_{i,j} - \hat{x}_{i,j}}{\sqrt{1 + h_{i,j}}} \tag{2.17}$$

With $h_{i,j}$ being the leverage:

$$h_{i,j} = x_{i,j-1}[X'_{i-1,j-1}X_{i-1,j-1}]^{-1}x'_{i,j-1} \tag{2.18}$$

A clear appeal of using the recursive residuals for diagnostics is that if there are any departures from the model assumptions, all of the residuals are affected by it [19], because all sensors are used in the model.

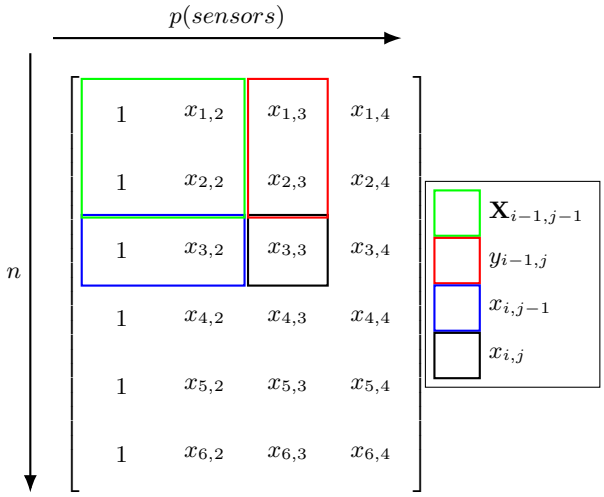


Figure 6: Example matrix of recursive residuals determination with least squares regression model.

Writing the recursive residuals from Equation 2.17 in a p by n matrix gives the first transformation step and the $R^{n\times p}$ matrix. Where the $r_{i,j}$ in R are N(0, $\sigma^2_{i,j}$), with σ^2 the conditional variance of $x_{i,j}$ given $x_{i,0}, ..., x_{i,j-1}$. Consider the following example: we want to predict $x_{3,3}$, see Figure 6. Then we first have to solve the following linear regression (via Cholesky decomposition):

$$y_{1:2,3} = X_{1:2,1:2}\beta_{2,3} + \epsilon_{3,3} \tag{2.19}$$

And find $\hat{x}$ using the obtained $\beta_{2,3}$:

$$\hat{x}_{3,3} = x_{3,1:2}\beta_{2,3} \tag{2.20}$$

In general, for finding $\hat{x}_{n,3}$, we would have two independent variables in column 1 and 2. Note that if we solve for $x_{3,2}$ we would get $\hat{x}_{3,2} = E(y_{1:2,2})$, which is simply the mean of the dependent variable column. This example highlights the importance of the order of sensors in the design matrix X. Hawkins notes that there is potential value in changing sensor order by putting alumina measurements first [7]. Sensors near the end columns of X are assumed to have more independent sensors, while the sensor at the first column is assumed to have no independent sensors. If the covariance matrix were known exactly, our prediction of $\hat{x}$ could be improved by ordering the sensors based on covariance, putting sensors with high summed covariance with the other sensors at the end columns of X for example.

Next the transformation to a multivariate standard normal distribution U in $N(0, I)$ can be made by studentizing the recursive residuals:

$$t_{i,j} = \frac{r_{i,j}}{\sqrt{\sum_{k=j+1}^{i-1} \frac{r^2_{k,j}}{i-j-1}}} \tag{2.21}$$

Based on $t_{i,j}$, $u_{i,j}$ can be defined according to:

$$u_{i,j} = \phi^{-1}[F_{i-j-1(t_{i,j})}] \tag{2.22}$$

Where ϕ^{-1} denotes the inverse normal distribution and F_{i-j-1} the cumulative distribution function of t. The vector $U_i = [u_{1,p}, ..., u_{i,p}]$ can be obtained by determining $u_{i,j}$ for all observations. U_i can now be monitored by any multivariate scheme. Hawkins used the MEWMA chart:

$$M_i = \lambda U_i + (1 - \lambda)M_{i-1} \tag{2.23}$$

Where $0 < \lambda \leq 1$ is a smoothing constant, M_i is defined for the ith observation and M_0 is 0. The MEWMA chart signals if the $||M||^2$ statistic exceeds the asymptotic control limit:

$$LIM_{i,j} = \frac{\lambda[1 - (1 - \lambda)^{2(i-j-1)}]}{2 - \lambda}h \tag{2.24}$$

Where h should be calculated numerically to specify the ARL_{IC}. Hawkins points out that the asymptote of Equation2.24 could also be used as the control limit, however this diminishes the chance of early detection of shifts. An example of the SSMEWMA for a run length of 30 observations and no shift is shown in Figure 7.

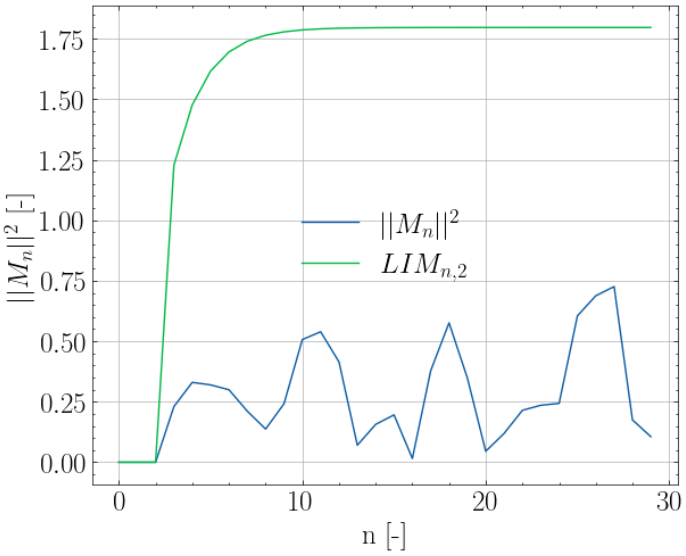


Figure 7: SSMEWMA control chart [7] example for a run length of 30 observations and no shift.

One downside of the SSMEWMA is that R is undefined for i < j, as the design matrix ($X_{i-1,j-1}$) used in the leverage calculation is not of full rank. The higher the covariance between sensors monitored by an SSMEWMA chart, the smaller the residual variance ($\epsilon_{i,j}$) of the model and the larger the shift in $||M||^2$. In general, the SSMEWMA chart is especially sensitive to small process shifts [7].

Process data could be correlated and contain many different sensors. However, direct monitoring with a single SSMEWMA chart is often impossible due to the large number of sensors exceeding the small number of observations. Therefore, we propose to use multiple groups of sensors monitored by SSMEWMA-charts simultaneously. The groups can be defined before starting the process or in real-time, where new groups are found after each new observation. In Section 2.5 we propose a procedure (P-SSMEWMA) for finding groups in real-time to be monitored by multiple SSMEWMA-charts.

2.5 P-SSMEWMA: group finding procedure

In order to apply the P-SSMEWMA procedure, a method for finding similar groups should be defined. In this section, we propose a new procedure for determining groups of sensors to be monitored by SSMEWMA charts. It is assumed that the number of groups in the data is unknown. We turn to step 2 of the procedure in Figure 8 and define a partitioning algorithm for finding similar groups. The proposed partitioning algorithm extends the use of the SSMEWMA chart to applications with many sensors and few observations. A Bayesian approach seems well-suited due to the small number of observations. In Section 2.5.1, an objective function is derived from a Bayesian perspective that scores the groups of sensors and is related to the POD. In Section 2.5.3 the procedure for finding groups is formulated as an optimization problem.

2.5.1 Bayesian perspective of recursive linear regression

Let $\{\mathcal{M}_i^{(j)}\}$ for each $j = 1, ..., M$ and $i = 1, ..., k$ denote multiple sets of different recursive linear regression models, with M all possible sets of models and k the number of models in set j. Each set j corresponds to a different assignment of sensors over models. $\{\mathcal{M}_i^{(j)}\}$ describes the full data set D_n at observation n. The exact model used is shown in Equations 2.15 and 2.16.

We define y_n as a scalar value for the state of sensor p at observation n. Note that $\hat{x}_n$ in Equation 2.16 is equal to $\hat{y}_n$ and is the estimation of the state by the model $\mathcal{M}_i$ using X_{n-1} as the design matrix containing all observations up until n-1 for all sensors 1 to p-1. For simplicity in notation, the subscript indicating the sensors is dropped. We use x_n to denote the nth row of observations. We now follow a derivation given by Chen [20] and assume that y_n follows a first-order Markov process:

$$p(y_n \mid y_{0:n-1}) = p(y_n \mid y_{n-1}),$$

In a review article about the recursive residuals [21] and earlier work by Hawkins [22], a real-time updating scheme for the regression coefficients (β) is stated as an alternative to batch updating for more efficient computations, here the assumption of a first-order Markov process is also made. Furthermore, we also assume that the observations x_n are independent of the states y_n, which is shown by Hawkins [7]. With these assumptions, the conditional probability density function (posterior probability) of y_n at observation n, given X_{n-1}, can be written as:

$$p(y_n|X_{n-1}) = \frac{p(x_n|y_n)p(y_{n-1}|X_{n-2})}{p(x_n|X_{n-1})} \tag{2.25}$$

Where $p(x_n|y_n)$ is the likelihood or the probability of the observation x_n given y_n. $p(y_{n-1}|X_{n-2})$ is the prior distribution or the posterior of the previous observation at n-1, since our model is recursive. $p(x_n|X_{n-1})$ is a normalization constant where the predictions y_n have been integrated out and is referred to as the marginal likelihood or evidence. The evidence does not depend on y_n, thus we can approximate the posterior as:

$$p(y_n|X_{n-1}) \propto p(x_n|y_n)p(y_{n-1}|X_{n-2}) \tag{2.26}$$

Now, $p(y_n|X_{n-1})$ gives the probability density function of the state y_n. See Figure 3, $P(X|nodefect)$ can be seen as the IC distribution for sensor p, which is equivalent to $p(y_n|X_{n-1})$. Evaluating $p(y_n|X_{n-1})$ at its maximum a posteriori (MAP) estimate (mode) gives the maximum value for the probability of the prediction. Increasing the probability at the MAP estimate leads to a sharper peak, decreasing the uncertainty in the prediction. Ultimately, this leads to less

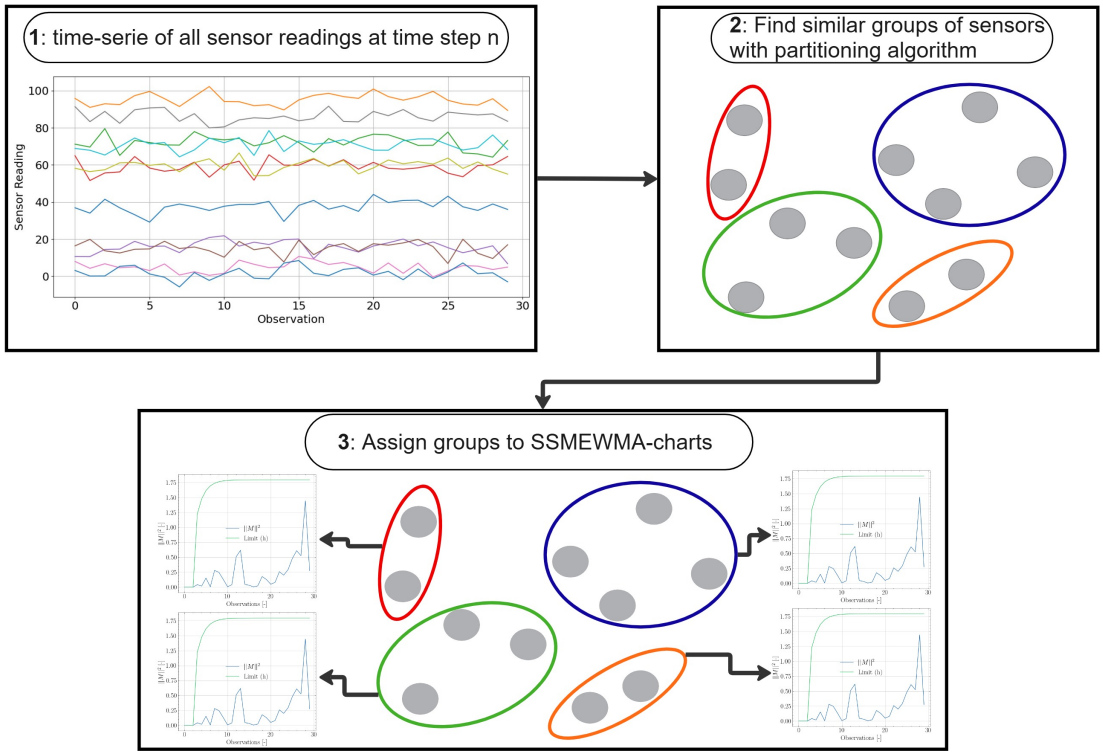


Figure 8: High level overview of P-SSMEWMA procedure for example grouping process data and monitoring with control charts.

overlap between the IC and OOC distributions (Figure 3) and consequently a higher POD. With this in mind we can formulate our objective for finding sensor groups as follows:

For n observations, find the set j that maximizes $p(y_n = y_{MAP}|X_n)$ across all models in the set:

$$\hat{\mathcal{M}}_i^{(j)} = \arg\max_{\mathcal{M}_i^{(j)}} \sum_{i=1}^{k} p_i(y_n = \hat{y}_{\text{MAP}} \mid X_{n-1}) \tag{2.27}$$

In the next section we will show how we define $p(y_n = \hat{y}_{\text{MAP}} \mid X_{n-1})$.

2.5.2 Posterior distribution

From Equation 2.26, we see that the posterior distribution is a function of the product between the likelihood and the prior. We assume that the likelihood and prior are conjugate-Gaussian (see chapter 2.3 of Bishop for more details [23]). The posterior distribution for a group of p sensors will then be a multivariate Gaussian and can be written as:

$$p(y_n \mid X_{n-1}) = \frac{\exp\left(-\frac{1}{2}(y_n - \beta_{n-1}X_{n-1})^T\Sigma_{n-1}^{-1}(y_n - \beta_{n-1}X_{n-1})\right)}{(2\pi)^{\frac{p}{2}}|\Sigma_{n-1}|^{\frac{1}{2}}} \tag{2.28}$$

With β and Σ_{n-1} now being given by the posterior updates for conjugate Gaussians, see [23] and [24] for more details. Evaluating Equation 2.28 at the maximum likelihood (MLE) estimate gives:

$$p_{MLE}(y_n = \hat{y}_{MLE} \mid X_n) = (2\pi)^{-\frac{p}{2}}|\hat{\Sigma}_{MLE}|^{-\frac{1}{2}} \tag{2.29}$$

Where $|\Sigma_{MLE}|$ denotes the determinant of the MLE estimated covariance matrix based on the data X_{n-1}. The maximum a posteriori estimate (MAP) could also be used, for multivariate Gaussians the MLE and MAP obtain the same values [23]. Using the MLE for covariance estimation is an approximation of the Bayesian approach to recursive parameter estimation.

For instance, Σ_{MAP} could be used in Equation 2.29 and calculated using the regularized MAP estimator from Murphy [24]:

$$\hat{\Sigma}_{MAP} = \lambda \times \Sigma_0 + (1 - \lambda) \times \hat{\Sigma}_{MLE} \tag{2.30}$$

Where Σ_0 is the prior for the covariance matrix and $\hat{\Sigma}_{MLE}$ the MLE covariance estimate. The regularization parameter λ is defined as: $\lambda = \frac{N_0}{N_0+N}$, where N_0 controls how much weight is put on the prior. This regularization estimator resembles Ledoit-Wolf shrinkage [25]. However, for a full Bayesian treatment based on model evidence we refer to Appendix A.1. Here, we show how Equation 2.29 can be derived with the model evidence.

For small samples, the Bayesian approach can prevent over fitting and improve numerical stability by using a prior. This is necessary, as the estimated covariance matrix in Equation 2.29 is unstable for small sample sizes. The prior Σ_0 could be specified in any desired way. For instance, a conservative prior could be set such that: $\Sigma_0 = \frac{I}{N_0}$, with I the identity matrix. If process knowledge is available, it is also possible to set the prior to reflect this knowledge.

2.5.3 Final objective and constraints

Using Equations 2.27 and 2.29 we can define a score function that can be used to compare different sets of models. The natural log of Equation 2.29 is taken for easier calculations and to remove extremely large and small values in the objective function. Lastly, we define the objective in terms of a loss function to minimize, similar to the BIC-cost [24]. We define the loss for a set j consisting of k models as:

$$\mathcal{L}_j = \sum_{i=1}^{k} \frac{p_i}{2} ln(2\pi) + \frac{1}{2} ln|\hat{\Sigma}_{MLE,i}| \tag{2.31}$$

Our goal then becomes to find different configurations of sensors in models that minimize Equation 2.31. The number of possible ways to partition p sensors into k disjoint non-empty groups is extremely large and can be calculated using Stirling numbers of the second kind [26]. Furthermore, this score function is non-linear and many local solutions might exist. Therefore, an exact solution is unfeasible and we will need to use heuristic based methods for generating solutions. The generated solutions are subject to one constraint; each sensor group should consist of at least 2 sensors.

To conclude, finding the set j that minimizes Equation 2.31 ensures that the POD of detecting a change in the found set is maximized. Meaning that by minimizing Equation 2.31, we can improve the POD of all models in the set. This procedure is an approximation of a full Bayesian treatment for parameter estimation and allows for fast computations.

2.6 Control charts summary

The Q-chart, SSMEWMA-chart and HC-chart were explained in the previous section. Here a quick overview is given of each chart:

- **Q-chart**: The Q-chart is a self-starting univariate chart that does not consider correlations between sensors. It is an intuitive chart as it monitors the original data with updating control limits.
- **SSMEWMA**: The SSMEWMA-chart is a self-starting multivariate chart that considers correlations between sensors. The SSMEWMA cannot monitor processes where the number of observations is less then the number of sensors. Therefore, for processes with many sensors it is necessary to create groups.
  - **R-SSMEWMA**: The R-SSMEWMA monitors groups of sensors that are assigned randomly.
  - **P-SSMEWMA**: The P-SSMEWMA monitors groups of sensors that are assigned based on the minimization of Equation 2.31.
- **HC-chart**: The HC-chart is a self-starting multivariate chart that can monitor processes even when the number of observations is less then the number of sensors in the chart.

3. METHODOLOGY

In this section, we describe the research design in three parts. First, in Section 3.1, we demonstrate how the control charts are implemented with a specific focus on the implementation of our proposed procedure. At the end of the section we show how the control limits are set. Second, in Section 3.2, we explain the experimental set-up used for the simulations based on the case study of the electron optical module 1. Third, in Section 3.3, several parameters to be varied for sensitivity analysis and tests to verify the proposed grouping procedure are discussed.

3.1 Implementation and control limits

The implementation of all control charts as well as the simulations are performed in Python. The code used for implementation, simulations and values for control limit settings can be found on Github. The simulation results are also shared. However, the data of the case study for optical module 1 is not shared. See Github page (https://github.com/ThijsBeene/Simulations). To ensure that the control charts are correctly implemented we performed verification simulations. These were performed for the SSMEWMA and the HC chart, as the implementation proved complex. The verification procedure consisted of reproducing results from the respective papers [7][17]. The results can be found in the Appendix.

3.1.1 Q-chart implementation

The Q-chart is implemented according to the theory in Section 2.2. Each Q-chart monitors a separate sensor, therefore we can directly apply the Q-chart to generated data without specifying groups.

3.1.2 R-SSMEWMA implementation

The R-SSMEWMA is implemented according to the theory in Section 2.4. However, it is assumed that we have no knowledge about what groups should be monitored. Therefore, for the R-SSMEWMA, sensors are randomly assigned to groups containing p sensors and these groups are monitored by SSMEWMA charts.

The group size p, is set to 2 sensors. This is done to have the maximum POD with respect to group size, see Equation 2.29. For random assignments, we can assume that $|\Sigma_{MAP}|$ is approximately constant across different value for p. Thus Equation 2.29 can be approximated by $p_{MAP} = (2\pi)^{-\frac{p}{2}}$. p_{MAP} is maximized when p is small, or in our case $p = 2$. Different values for p are tested in the sensitivity analysis to verify this hypothesis.

3.1.3 HC-chart implementation

The HC-chart is implemented according to the theory in Section 2.3. One HC-chart can monitor all sensors with a derived statistic, therefore we can directly apply it to the generated data without defining groups.

3.1.4 P-SSMEWMA implementation

The P-SSMEWMA is implemented according to the theory in Sections 2.4 and 2.5. Equation 2.31 is minimized with $\Sigma_{MAP} = \Sigma_{MLE}$. The problem of finding groups of sensors by minimizing Equation 2.31 will be considered in the context of graph theory. Where we have a graph G(E, V), with E the edges and V the vertices. The vertices represent sensors, while the edges represent the covariance between vertices.

A maximum spanning tree (MST) is constructed using Kruskals algorithm. Starting from the MST, groups can be found by changing edges and evaluating the score function (Equation 2.31). This procedure is similar to the classic single linkage scheme [27]. Contrary to the single linkage scheme, where edges are only removed, we swap edges for edges not in the current solution.

The procedure starts by randomly choosing an edge from the current solution, with the initial solution being the MST. Next, an edge is selected from the full set of possible edges that is not in the current solution and has a weight higher then the edge to be replaced. The new solution that now includes the swapped edge is checked for viability by verifying that every node is connected to at least one other node.

If the check succeeds, the value of the objective function of the new solution is computed using Equation 2.31. The new solution is accepted as the current solution if the score is lower then the previous solutions score. These steps are repeated until the current solution does not change much. Finally, group labels can be found by assigning the same label to nodes that are connected. See Algorithm 1 for the pseudo-code of a greedy implementation of the random edge swapping procedure.

Algorithm 1 is a greedy algorithm that finds groups of sensors by iteratively removing the lowest covariance edge and adding the highest covariance edge that is not in the current solution. If the current score is improved the solution is always accepted. However, the algorithm can easily get stuck in local optimal solutions, due to the inability to explore solutions that initially lead to a worse configuration. Therefore, we supplement the algorithm with two other heuristic based algorithms: simulated annealing (SA) and Tabu search.

Algorithm 1 Greedy random edge swapping algorithm for finding P-SSMEWMA labels

Require: MST edges (G(V,E)), all edges
Ensure: Sets: `ssmewma_labels`, `q_labels`
Set current edges <- MST
Set best cost <- Cost(current edges)
for each iteration **do**
  for each attempt **do**
    Edge to remove <- select lowest(current edges)
    Edge to add <- select highest(all edges > weight(edge to remove))
    Proposed neighbor <- current edges | Edge to remove and edge to add
    if Proposed neighbor maintains connectivity (all d > 0) **then**
      break
    end if
  end for
  Compute cost of proposed neighbor
  if cost < best cost **then**
    Update best solution
    Best cost = cost
    current edges = proposed neighbor
  end if
end for
Final labels <- Labels(current edges)

3.1.5 Heuristic algorithms

The SA algorithm decreases a parameter T, related to the temperature in an actual annealing process. This parameter influences the probability of accepting a worse solution as the current solution, with $P(accept) = e^{\frac{\Delta}{T}}$, where Δ is the difference between the current and proposed solution. By adding this probability of accepting to Algorithm 1, we obtain a broader exploration of the solution space.

The Tabu search algorithm is similar to the SA algorithm in that it allows for the exploration of local worse solutions. The algorithm always selects the best neighbor solution, even if it has a worse score then the current solution. Each new best neighbor solution is added to a list of solutions that the algorithm can not revisit.

The three algorithms (greedy, SA and Tabu) are tested on the optical module 1 data set. All three algorithms are set to run for an amount of iterations to find the best solution. The found solution for all algorithms is evaluated after 5 seconds. This is a short, yet necessary time frame, as the application of the algorithms in real-time SPM requires fast computations. For the simulations in this research, we consider a time of 1 second to find the best solution, as overall simulation times can drastically increase due to the grouping algorithm. In total, 50 runs are performed to account for the randomness in the SA and Tabu algorithms, the 95% CI is shown. Based on the objective function score after 1 second of run time, we select the algorithm for our simulations.

3.1.6 Control limit setting

For a fair comparison between charts, the ARL_{IC} was set to 25 observations. The high-tech workcenter is expected to produce approximately 25 modules per year. Having one expected false signal per year is considered appropriate for this application.

All limits were determined numerically, to circumvent making assumptions about the underlying data. We start by generating random IC data based on the MLE estimates of the optical module 1 data. 100 runs are generated with a run length of 200 observations. Next, the value for the control limit that results in an ARL_{IC} of 25 observations is determined using the Brent algorithm [28].

For the Q, R-SSMEWMA and HC-charts, calculating the limits is straightforward. However, for the P-SSMEWMA the calculation is more involved. This is because in a single run we have several charts monitoring a varying number of sensors. Each control chart has a different limit depending on the number of sensors monitored and the total number of charts. Therefore, we determine the value for h similar to the approach by Hawkins et al [7]. However, we now need to find h for all combinations of group size and number of groups. This provides us with a matrix of h values that result in the correct $ARL_{IC,O}$. The numerically found values for the control limits for the four charts are shown in the Appendix A. The control limits for the P-SSMEWMA can be found in the Python code.

3.2 Case study: electron optical module 1

The control charts are applied to the case study of the electron optical module 1. The MLE estimates are obtained from the dataset, see Section 1.1.1. 200 simulated process runs are generated based on these MLE estimates. The generated runs are then processed for use by the different control charts. For the Q and HC charts, the data can immediately be given as input and we do not have to determine groups. For the P and R-SSMEWMA it is necessary to find groups of sensors in the data. This is done as described in Section 3.1. The following data formats are given as input to the different control charts:

1. **Q-chart**
   (a) Data input: full data set without groups. Each Q-chart monitors a separate sensor.
2. **R-SSMEWMA**
   (a) Data input: 48 random groups each of 2 sensors.
3. **P-SSMEWMA**
   (a) Data input: groups of varying size determined with partitioning procedure
4. **HC chart**
   (a) Data input: full data set without groups. One HC-chart monitors all sensors.

In total, four simulations will be performed. These four simulations correspond to different settings of the parameters n and τ: $n \in [15, 20, 30, 40]$ and $\tau \in [5, 10, 20, 30]$. For each simulation, a persistent mean shift is induced at observation τ of magnitude $\delta \in [1, 2, 3, 4, 5]$ standard deviations for a randomly selected sensor (p). We use Equation 2.2 to calculate the delay in detecting the OOC event for each chart.

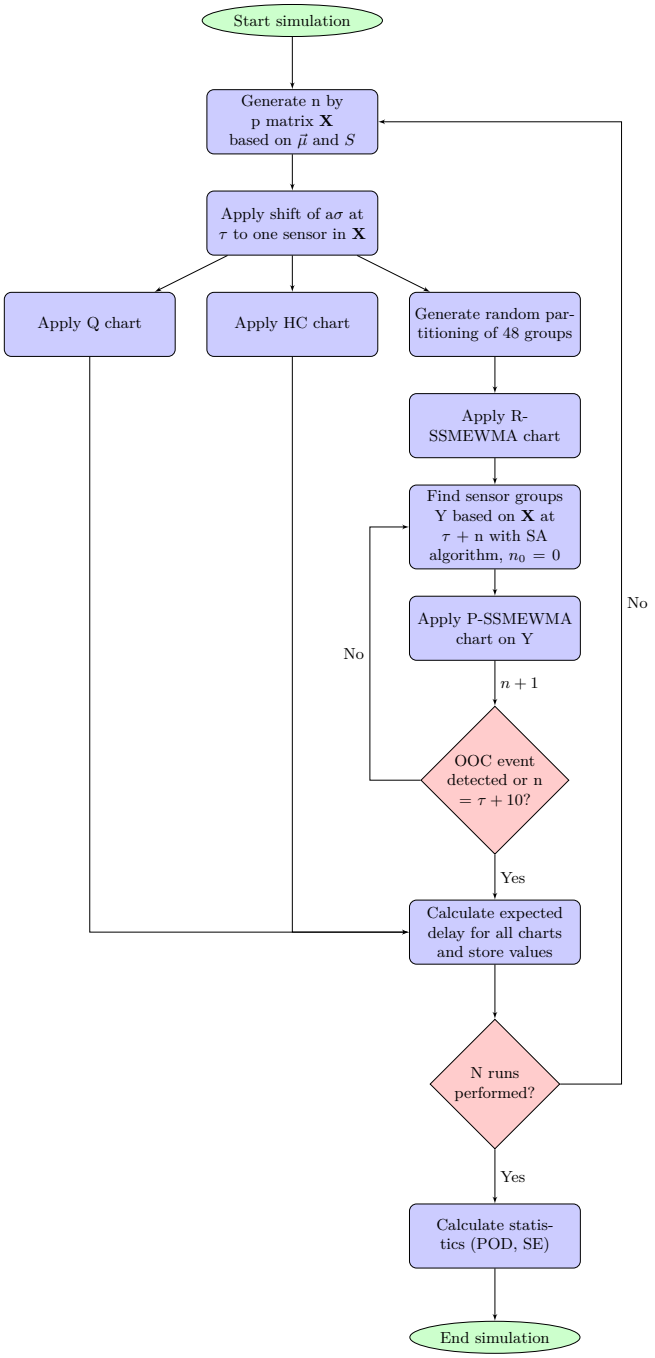


Figure 9: Experimental set-up for case study simulations in flowchart

For all charts, we iterate over the observations, at observation τ, we start checking if the chart monitoring p signals an OOC event. If no signal is given we keep iterating until we reach the final observation n, which is always set to be τ+10. If no signal is given, the expected delay for that run is set to

zero. This procedure is repeated for 200 randomly generated runs, all charts are applied to the same generated random data. The POD is calculated of early detection (2 observations), medium detection (5 observations) and late detection (10) observations. See Figure 9 for a schematic overview of the simulation. An overview of the parameter settings during the simulation is shown in Table 1. Some simulation parameters remain constant throughout the whole simulation, these are shown in Table 1.

Table 1. General simulation settings

N	100
Max r	10
λ (SSMEWMA)	0.25
$ARL_{IC,O}$ all charts	25
SSEWMA cluster size	2
Total sensors	96
Actual number of observations	30
Initial temperature SA	1
Cooling rate SA	0.99
Maximum iterations SA	1000
Minimum temperature SA	$1e^{-5}$
Neighbor solution attempts SA	20
Neighbor solution attempts Tabu	20
Maximum iterations Tabu	100
List size Tabu	50
Max iterations Greedy	50
Top candidates considered for adding Greedy	20

3.3 Sensitivity analysis

3.3.1 Group size and POD

The R-SSMEWMA is defined as multiple SSMEWMA charts monitoring groups of randomly assigned sensors, where each group has an equal number (p) of sensors. This forces p to take on values that can exactly divide the total number of sensors (P), otherwise we would have remaining sensors. Here, p is considered a parameter that can be varied to obtain a desired POD.

The POD for a fixed value for p is calculated similar to the procedure outlined in 9. A shift of $\delta = 4$ is induced at observation 20 and the POD is calculated after 10 observations. We iterate over various values for p: $p \in [2, 3, 4, 6, 8, 12, 16, 24]$ and determine the POD for each value of p. In total, 1000 runs per p value are performed.

3.3.2 Objective function verification

To show the relation between the POD and the derived loss function (Equation 2.31), we set-up an experiment where we generate multiple runs for p sensors monitored by a single SSMEWMA chart and varying p for: $p \in [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$. At observation 20 we induce a shift of $\delta = 10$ σ. At observation 20 we calculate the value of the loss function Equation (2.31) and the corresponding relative shift in the SSMEWMA chart ($||M_n||^2_{rel}$), which is defined as: $\frac{||M_n||^2}{LIM}$. $||M_n||^2_{rel}$ is proportional to the POD. It is expected to see a relation between these quantities, as shown in Equation 2.29.

Furthermore, δ is varied from 2 to 16 σ to investigate how well the relation holds up for different shifts. Lastly, τ is also varied from 10 to 45 to see what the effect of an early or late process shift is. Due to the regularized covariance, it is expected that the relation will be more pronounced for larger τ.

4. RESULTS AND DISCUSSION

In this section we show the results. First, the performance of the grouping algorithm is investigated. Next, the results for the four control charts (Q, R-SSMEWMA, P-SSMEWMA and HC) applied to the case study are shown. Lastly, a sensitivity analysis is performed to look into the relation between the POD and loss function.

4.1 Partitioning algorithm performance

In Section 3.1.4 we defined a loss function (Equation 2.31) to determine groups of sensors. The SA, Tabu and greedy algorithms are all used to minimize Equation 2.31.

4.1.1 Fixed observations

First, we apply the three algorithms to a fixed number (30) of observation. The maximum spanning tree (MST) is found using Kruskal's algorithm, this is shown in Figure 10. All MST nodes are connected and there exists no cycles in the graph. This MST is the initial solution for the minimization problem and used as input for the SA, Tabu and Greedy algorithms, see Algorithm 1.

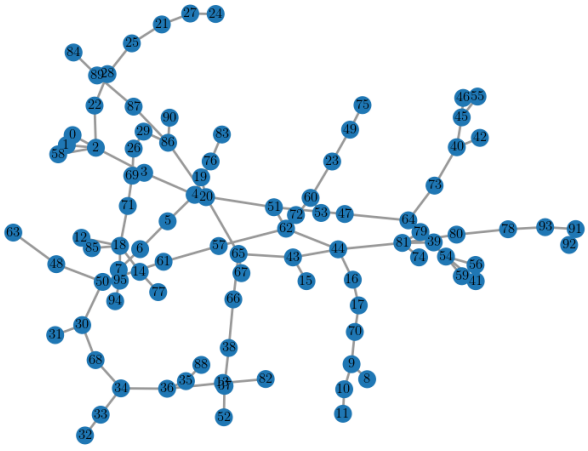


Figure 10: Maximum spanning tree of electro-optical module 1 data. Nodes represent sensors and edges represent covariance between sensors.

Next, the SA, Tabu and greedy algorithms are applied to the MST in Figure 10. The lowest found score of the loss function for the three algorithms is shown for the first 5 seconds. The run time is limited to 5 seconds, as this is considered to still be manageable for real-time SPM. Increasing the run time could still improve the solution, however computational times should also be considered.

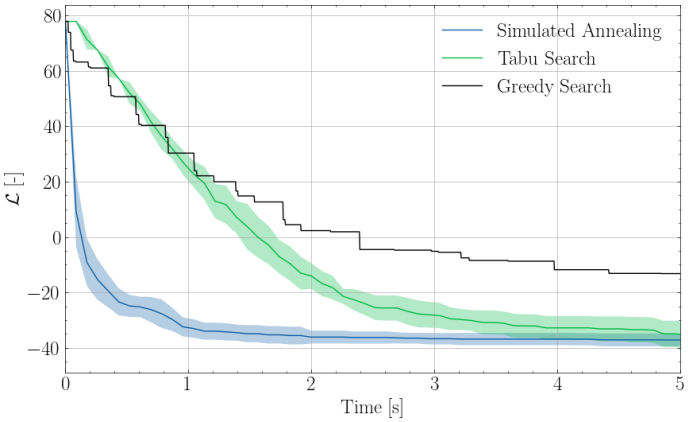


Figure 11: Value of Equation 2.31 over 5 seconds for the Greedy, Tabu and SA algorithms. The Tabu and SA algorithms are not deterministic, therefore the mean with 95% CI is shown over 10 runs of the algorithms. In total 30 observations were used. Nodes represent sensors and edges represent covariance between sensors.

From Figure 11 it can be seen that the SA algorithm managed to find solutions with the lowest score for the loss function after 1 second and after 5 seconds. The found score after 5 seconds was on average -37.80. The Tabu algorithm performs slightly worse compared to the SA algorithm, however it finds a slightly better solution after 5 seconds with an average score of -38.52. The Greedy algorithm performed the worst out of the three algorithms and found a solution with a score of -29.3924 after 5 seconds. The Greedy algorithm is relatively slow, since it can take many iterations before a new solution with a lower score is found. Allowing for a longer run time would likely improve the solution found by the algorithms.

For the simulations performed in this research, short run times are vital. This is mainly due to the application of the algorithm at each time step. A slight increase in run time can prolong the full experiment drastically. For this reason it was decided to use the SA algorithm, since it finds the lowest score solution after 1 second. Figure 12 shows the solution found by the SA algorithm after 1 second.

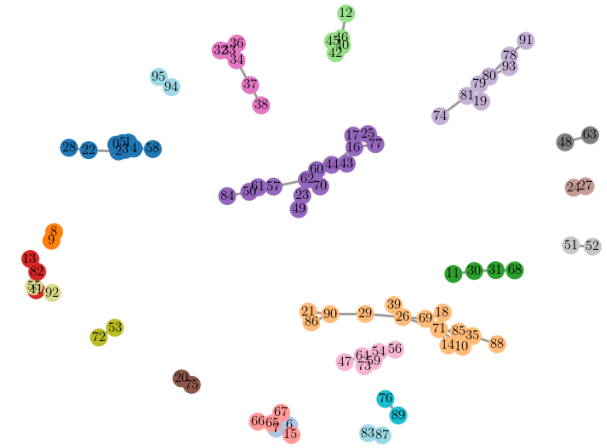


Figure 12: Graph of minimized $\mathcal{L}$ partition found with the SA algorithm after 5 seconds. Based on data for 30 observations of optical module 1.

The solution in Figure 12 is based on the original optical module 1 data with 32 observations and 96 sensors. It is noticeable that there do not seem to be many groups of average size. For example, we see quite a few groups with small sizes (2) and a few large groups (> 6), but not many averaged sized groups. This could reflect underlying relations between sensors or be attributed to the SA algorithm.

The SA algorithm swaps edges at each iteration of the algorithm. The main driver behind finding solutions is the removal of edges, since this process is ultimately what creates the groups of sensors. Forcing the adding of an edge as well creates a tendency to add edges to groups that are already large, because the influence on the loss is smaller when adding to an already large group.

Furthermore, the node and edge structure could potentially also reveal information about the process. For instance, in Figure 12 sensors 32, 34, 36, 37 and 38 belong to the same group, yet are all connected via sensor 34.

Visually, the found groups seem to reflect covariance between sensors. Since the sensors occupying the same group share high covariance among each other. To highlight this result, Figure 13 shows the correlations between sensors in a heat map for the first 21 sensors. Colors denoting the groups to which the sensors were assigned are shown on the sides of the heatmap.

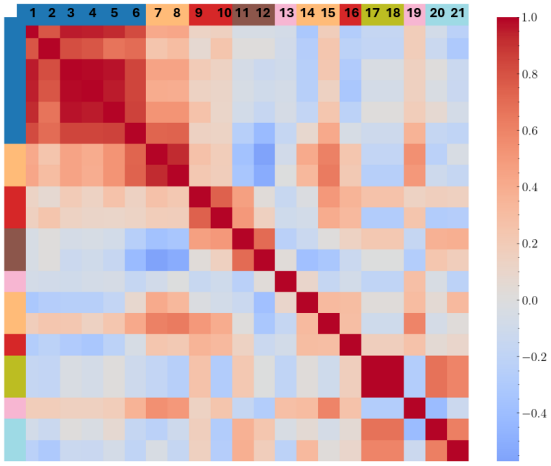


Figure 13: Subset of sample correlation matrix visualized with heatmap for 21 sensors. Colours along the axis denote the group to which the sensor is assigned. 30 observations were used to estimate correlation.

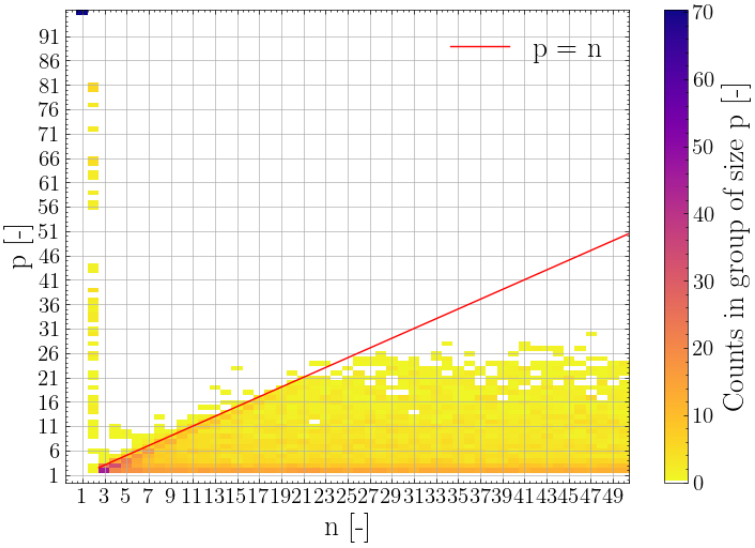


Figure 14: Average number of sensors per group of size p for 50 observations. In total 30 runs were performed, the average counts across these runs was taken.

From Figure 13 we can see that the groups are based on the covariance between sensors. For instance, the dark blue group contains the sensors 1,2,3,4,5 and 6. These sensors are all related to a crucial step in the work center, namely alignment during the stacking of MEMS layers. Therefore, from a process perspective, it is logical that they occupy the same cluster. Furthermore, sensors 7 and 8 also share high correlation with the dark blue group. However, they are also strongly correlated with each other, making the summed score lower by placing them in a separate group.

4.1.2 Varying observations

The estimated correlation matrix in Figure 13 is based on a fixed 30 observations. However, for a start-up process, the groups are found at each time step. To see what the configuration of groups is for a varying number of observations, see Figure 14.

The vertical bar denotes the average number of sensors in a group of size p. The red line shows where p is equal to n. The SSMEWMA cannot detect change if p exceeds n. Thus changes in groups above the red line will never be detected. Interestingly, the partitioning algorithm finds groups with sizes less then n (below the p = n line) most of the time. Starting from only 2 observations, most groups fall under the $p = n$ line. We emphasize that no constraint was placed on the maximum group size. The tendency to find groups below the $p = n$ line is likely coincidental and can be attributed to the fact that early MLE estimates of covariance are not well-defined. For example, when few observations are present, the MLE estimated covariance tends to by too high.

Since the algorithm automatically finds groups below the threshold, it is not necessary to place a constraint on the maximum group size. A few groups are above the threshold (see Figure 14), meaning that a change in any of these groups will never be detected. This slightly lowers the final POD. Instead of a constraint, an uninformative prior could also be used to force smaller group sizes. For instance, if an identity prior is used, Equation 2.29 only depends on the minimization of p, and would therefore assign all sensors to groups of 2 initially. As more observations come in the covariance estimate depends more and more on the data, and the groups sizes are allowed to increase.

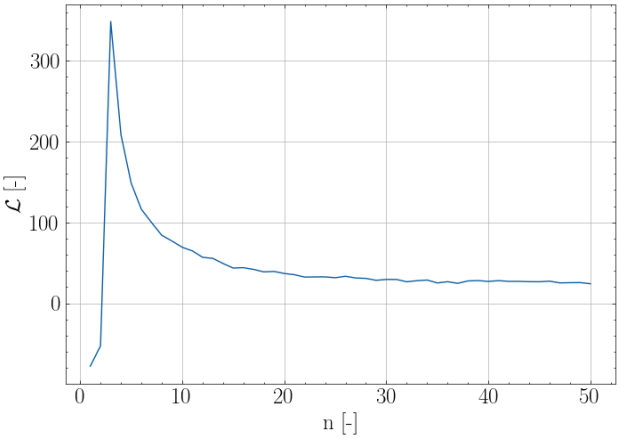


Figure 15: Average final value of $\mathcal{L}$ over 30 runs with 50 observations after applying the SA algorithm.

After a number of observations the found groups do not change much anymore. This already seems to be the

case after 20-30 observations. For practical implementation, it could therefore be advised to stop applying the P-SSMEWMA procedure and use the last partition of groups. Finally, see Figures 15 for the value of $\mathcal{L}$ for n observations and Figure 16 for a cross section of the counts from Figure 14

In Figure 15, the final value for $\mathcal{L}$ found by the SA algorithm decreases as n increases. This indicates that the found groups obtain a higher score for a more certain MLE estimate of covariance. This also corresponds to what is observed in Figure 14. When the final value for the loss function is high, the algorithm tends to find small groups of sensors. Small groups of sensors make the least assumptions about covariance between sensors (because we predict with neigbour sensors, see Section 2.4).

For the first 2 observations, the score appears to be lower. However, this is due to the MLE estimate of covariance not being well-defined. For 1 observations, an estimate cannot be made and all sensors are kept in 1 group. For two observations, the covariance among sensors is extremely large, which also leads to large groups and a low score.

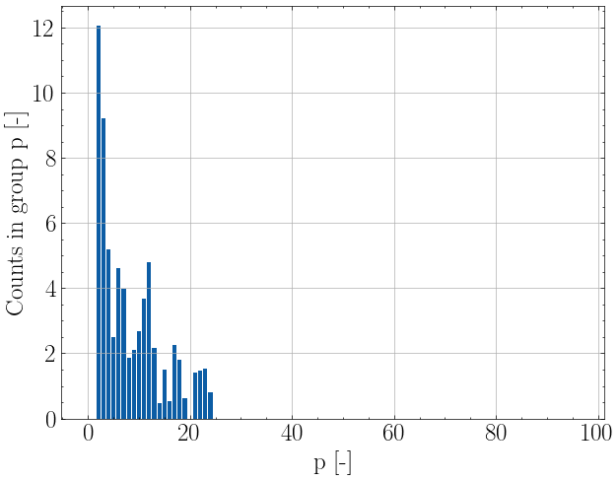


Figure 16: Cross section of average number of sensors in group of size p at n is 50 observations from Figure 14. Averaged over 30 runs.

Figure 16 shows the average number of sensors in a group of size p calculated at 50 observations. For instance, on average there are 12 sensors being monitored in groups of 2. Minimizing Equation 2.31 leads to most sensors being monitored in small groups. This is desirable, as small groups in general have a higher POD comapred to larger groups. However, sometimes large groups do form. This highlights that the POD is also improved when covariance between sensors is high. For large groups with high generalized variance, the prediction of the next observation for sensor i is more accurate due to the sensors sharing information. The probability of finding sensor i in a certain state given the states of all other sensors is higher when generalized variance is high.

4.2 Case study: optical module 1

In this section, the four control charts are applied to data generated based on the optical module 1. First, we verify that the ARL_{IC} of all charts is set to 25 observations. Next, we show the POD for a variety of situations of all charts applied to the optical module 1 data.

4.2.1 ARL_{IC} verification

Before a fair comparison between the four procedures monitoring the optical module 1 data can be made, we need to verify that the ARL_{IC} of all charts is equal. An ARL_{IC} of 25 observations was selected. Numerical simulations were performed to find the control limits corresponding to this ARL. The found limits are shown in Table 5 in the Appendix.

To verify that the found limits actually correspond to the desired ARL_{IC}, we performed a simulation with the found limits. See Figure 17 for the ARL_{IC} of the charts in a box plot showing the mean, 95 % confidence interval and minimum and maximum values.

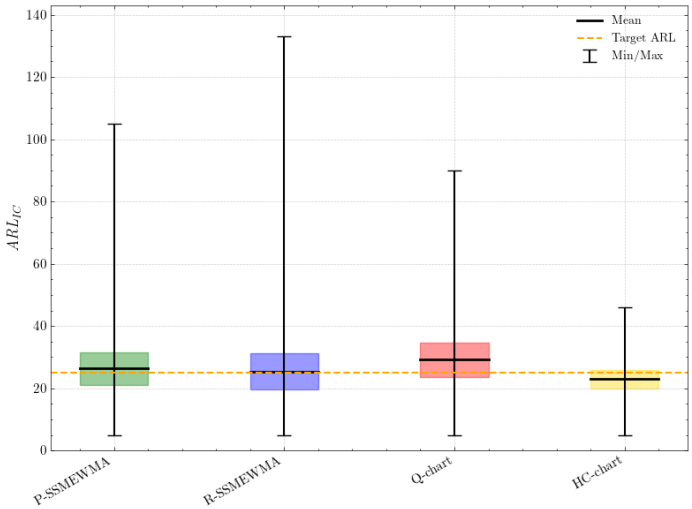


Figure 17: ARL_{IC} in a box plot for all four control chart procedures with numerically simulated control limits. Target ARL_{IC} was set to 25 observations. In total 25 runs were performed with a run length of 200 observations. The run length for the HC chart was set to 50 observations due to extremely long computational times.

Simulating more runs with a higher number of observations would improve our estimate of the limits, however Figure 17 shows that the target ARL_{IC} is within the 95 % confidence interval for all charts. Therefore, we conclude that our control limits are set correctly for comparison between charts. The run length for the HC-chart was shortened to only 50 observations due to computational times increasing drastically. This explains why the 95 % confidence interval is shorter and the maximum value lower.

4.3 Simulation results

The POD of all four control charts applied to the generated optical module 1 data for a shift δ at observation τ is shown in Table 2. From Table 2 we can see that the POD of the P-SSMEWMA is higher compared to the other charts for most combinations of τ and δ. Except for extremely early shifts, here the Hc-chart seems to have the best performance.

The performance of all charts deteriorates quickly for early shifts. For instance, shift at $\tau = 5$ are almost always missed, regardless of δ. This is to be expected, as the charts do not yet have a good estimation of the IC process. The POD for the HC-chart appears relatively high for early shifts. This could be due to a wrong setting of the ARL_{IC}, which was later solved and implemented in Figure 18. The POD of the HC-chart therefore likely contains a small constant positive bias in Table 2.

The Q-chart appears to have the worst performance for detecting early shifts out of all charts ($\tau = 5$). For fast detection of shifts, see the POD_2 column in Table 2, the Q-chart is well suited. There does not seem to be much difference between the POD_2, POD_5 and POD_10 of the Q-chart. This means that the Q-chart detects process shifts fast or does not detect shifts at all. The reason for this is that the Q-chart updates the control limits with the new OOC observations. If the shift is not detected fast, the control limits will have shifted towards the OOC distribution, making detection of the persistent mean shift more difficult.

The P-SSMEWMA manages to have a higher POD for smaller values of δ compared to the other charts. This is likely due to the SSMEWMA charts monitoring groups of sensors that were found by maximizing the probability at the MLE estimate. Any deviations, however small will be more pronounced if due to this maximization. It was expected that the chart could possibly become too sensitive to shifts due to the maximization, resulting in an increased false alarm rate that offsets the improvement sensitivity. However, this does not appear to be the case. For an easy visual comparison between control charts, we plotted the results for N = 30, $\tau = 20$ in Figure 18.

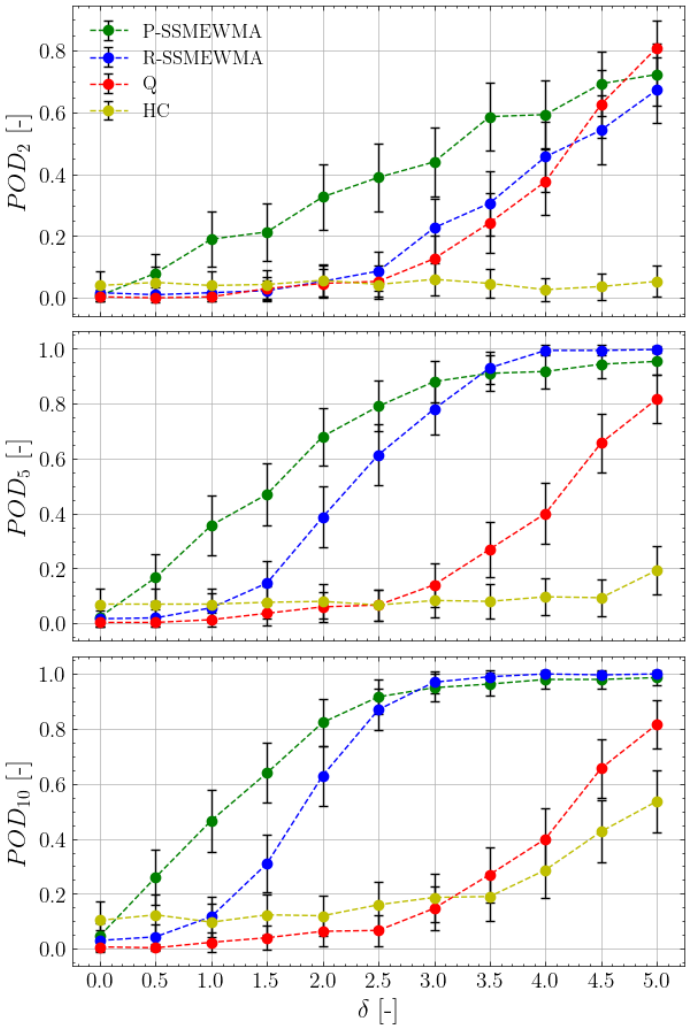


Figure 18: POD for all four control charts over a range of process shifts δ. τ was set to 20 observations and the total run length was 30. Error bars represent the 95% confidence interval. In total, 200 runs were performed and the shift was changed to include 0 and increment with steps of 0.5. The implementation of the HC-chart was slightly altered in this figure to ensure a 0 POD for a 0 shift.

In Figure 18 it can be seen that the POD is approximately equal to zero for the charts when the shift is equal to 0 σ. This is a result of the ARL_{IC} being set to 25 observations. The P-SSMEWMA yields a better POD_2 for all values of δ compared to the R-SSMEWMA. For instance, the POD_2 of the R-SSMEWMA already has a 20% chance of detecting a 1 σ shift, while the R-SSMEWMA is approximately at 0%. The P-SSMEWMA charts are more sensitive to initial shifts, due to the maximization of the posterior peak. This makes deviations from the estimated distribution stand out more, which increases the POD_2.

If the shift is not detected immediately, the shifted sensor observations are used in the covariance estimates. This slightly decreases the long run POD (POD_5 and POD_{10}),

Table 2. Results from simulation based on the case study for optical module 1. Various shifts δ were simulated for different τ. The best POD values are indicated in boldface. In total 100 runs were performed for each combination of δ and τ.

			Q-chart			R-SSMEWMA			HC-chart			P-SSMEWMA		
N	τ	δ	POD_2	POD_5	POD_{10}	POD_2	POD_5	POD_{10}	POD_2	POD_5	POD_{10}	POD_2	POD_5	POD_{10}
15	5	1	0	0	0.01	0	0	0	0.01	0.07	**0.19**	0.01	0.07	0.1
		2	0	0	0	0	0.01	0.03	0.01	0.03	0.09	0.02	0.05	**0.1**
		3	0	0	0	0	0	0.01	0.01	0.04	**0.21**	0.06	0.07	0.11
		4	0	0	0	0.01	0.01	0.01	0	0.07	**0.19**	0.06	0.09	0.12
		5	0	0	0	0	0.01	0.01	0.01	0.12	**0.36**	0.06	0.09	0.12
20	10	1	0	0	0	0	0	0.02	0.11	0.19	**0.34**	0.07	0.17	0.22
		2	0	0	0	0	0.03	0.06	0.05	0.12	0.27	0.12	0.29	**0.4**
		3	0.03	0.04	0.04	0.01	0.08	0.09	0.01	0.04	0.21	0.25	0.42	**0.51**
		4	0.06	0.06	0.06	0.07	0.19	0.31	0.04	0.19	0.51	0.26	0.49	**0.62**
		5	0.21	0.21	0.21	0.08	0.27	0.34	0.09	0.42	**0.75**	0.4	0.63	0.73
30	20	1	0	0.02	0.03	0	0.03	0.03	0.17	0.22	0.3	0.15	0.37	**0.53**
		2	0.01	0.02	0.02	0.01	0.15	0.27	0.15	0.25	0.33	0.24	0.68	**0.87**
		3	0.11	0.11	0.11	0.09	0.59	0.77	0.14	0.25	0.55	0.51	0.9	**0.94**
		4	0.39	0.39	0.39	0.3	0.91	0.98	0.15	0.39	0.72	0.66	0.97	**0.99**
		5	0.75	0.76	0.76	0.46	0.99	**1.0**	0.09	0.49	0.88	0.73	0.94	0.98
40	30	1	0	0	0.01	0.01	0.06	0.13	0.29	0.35	0.49	0.22	0.49	**0.67**
		2	0.03	0.05	0.07	0.03	0.2	0.61	0.2	0.27	0.49	0.4	0.84	**0.96**
		3	0.2	0.23	0.25	0.09	0.83	0.98	0.32	0.44	0.71	0.69	0.98	**1**
		4	0.57	0.63	0.64	0.4	0.99	**1**	0.22	0.44	0.85	0.87	1	**1**
		5	0.92	0.9	0.92	0.71	1	**1**	0.18	0.55	0.94	0.91	1	**1**

as groups are no longer based on the IC covariance estimate. To avoid this issue, a covariance estimator that is more robust to outliers could potentially be used. We can see in Figure 18 that the POD of the R-SSMEWMA improves drastically in the long run and is even higher compared to the P-SSMEWMA for $\delta > 3$. For smaller values of δ, the P-SSMEWMA still has a significantly higher POD then the R-SSMEWMA.

4.4 Sensitivity analysis

In this Section we perform a sensitivity analysis to determine the influence of different parameter settings. First, the group size (p) is varied to see how the POD changes for the generated optical module 1 data. Next, the value of the loss function Equation 2.31 is varied to see how the POD changes. Instead of directly using the POD, the relative shift is used, which is directly proportional to the POD.

4.4.1 R-SSMEWMA partition

To determine the maximum POD group size setting for the R-SSMEWMA we iterated over a range of group sizes. See figure 19 for the POD of various values of p with randomly assigned sensors.

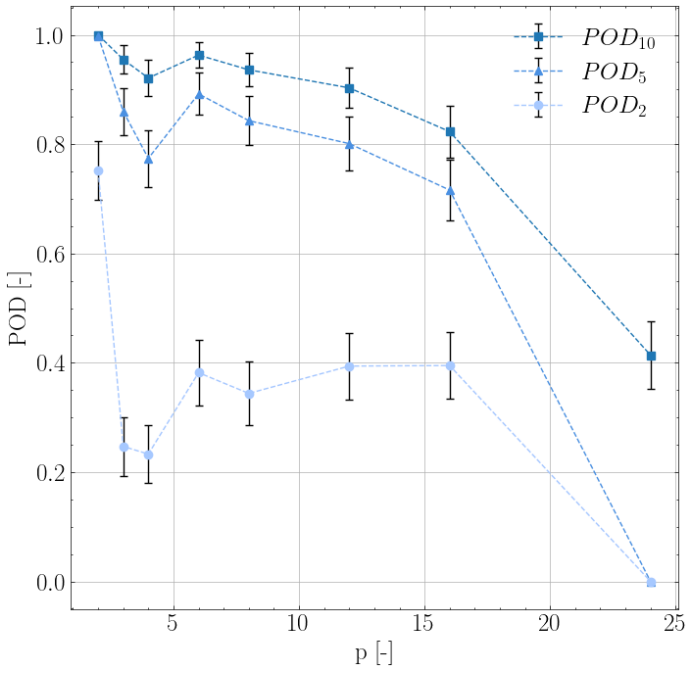


Figure 19: POD for various values of p. 2000 runs with 30 observations each were performed.

From Figure 19 it becomes clear that the POD is a function of p. As expected, a decrease in cluster size results in an increase in POD. This can also be seen in Equation 2.29. The maximum value for all POD is found for the smallest group size of $p = 2$.

4.4.2 Relation POD and $\mathcal{L}$

Investigating the relation between the POD and $\mathcal{L}$ is important because groups are determined based on the minimization of $\mathcal{L}$. We expect the POD (we will use POD and $||M_n||^2_{rel}$ interchangeably) to increase when $\mathcal{L}$ is minimized, see Section 2.5. To test if this relation is valid, several random covariance matrices of dimension p by p and corresponding data were generated. At observation 20, a shift was induced and $\mathcal{L}$ and $||M_n||^2_{rel}$ were calculated. Note that this is the initial shift at observation 20. The results are shown in Figure 20,

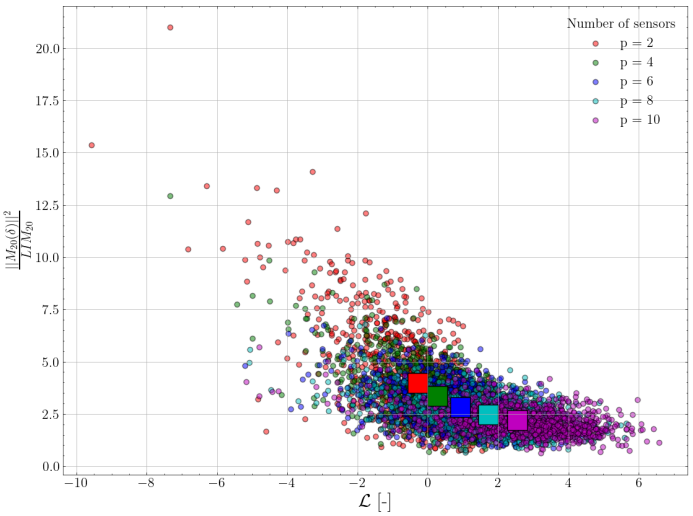


Figure 20: Relation between $\mathcal{L}$ and $||M_n||^2_{rel}$. For a shift of 10 σ at observation 20. Coloured squares represent the mean. In total 10000 runs were performed. Each scatter point represends a run based on data from a randomly generated covariance matrix.

From Figure 20 it can be seen that decreasing $\mathcal{L}$ results in an increase in POD. Furthermore, the relation seems to be independent of p, since the scatter points overlap. This is as expected and verifies the effectiveness of the used loss function. Figure 21 shows only the mean values of $\mathcal{L}$ and $||M_n||^2_{rel}$, which are denoted with a bar.

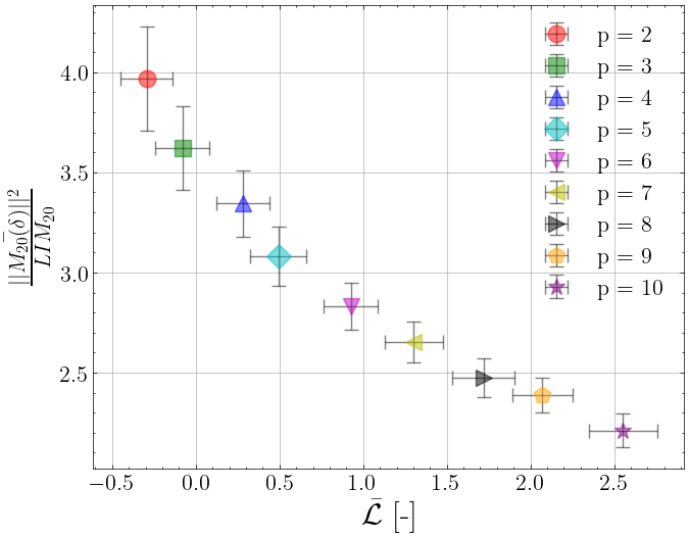


Figure 21: Relation between $\mathcal{L}$ and $||M_n||^2_{rel}$ with only the mean shown. For a shift of 10 σ at observation 20. Error bars represend the 95% CI.

The mean values in Figure 21 appear to follow an exponential relation. Now we investigate how the results in Figure 21 hold up for different values of τ and δ. We use different markers to distinguish between different sensor sizes, but do not explicitly state this in the graph. First, we look at the effect of changing δ. This is done by repeating the simulations for Figure 21, but for δ ranging from 0 to 16. The results are shown in Figure 22.

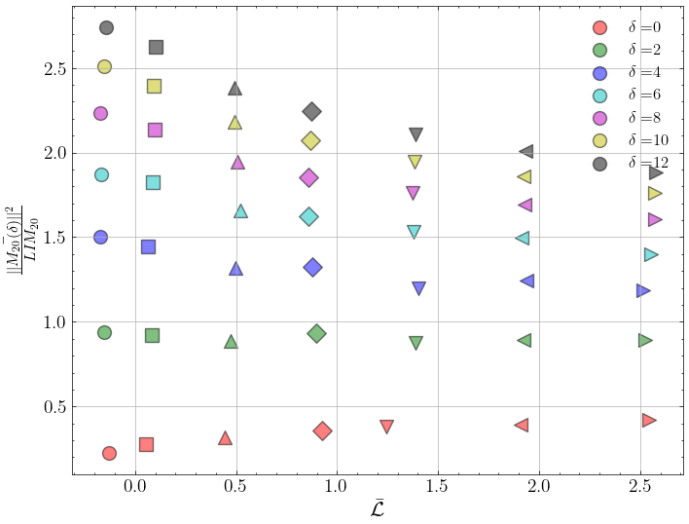


Figure 22: Relative shift for several shifts in the acutal data of different magnitudes. τ was kept constant at 20 observations. Markers denote groups of similar size. Mean values are plotted. In total 3000 runs were performed. The left most marker is for p = 2, while the right most marker is p = 10.

From Figure 22 it can be seen that the relation between POD and $\mathcal{L}$ holds up relatively well for different values of δ. For $\delta = 0$, the relation is approximately constant, which is as expected, since it would be undesirable to have an increase in false positives due to the grouping procedure. For small shifts, the relation is less pronounced, but still existent. τ was kept fixed at 20 observations.

Now we look into the effect of a change in τ on the POD. δ is fixed at a value of 10 σ and τ is varied from 10 to 45 observations. The results are shown in Figure 23.

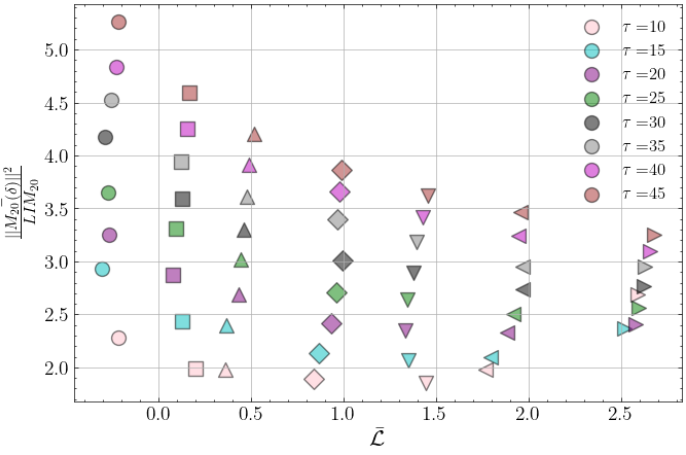


Figure 23: In total 3000 runs were performed, markers represent group sizes. Shift magnitude was kept constant at 10 σ. The left most marker is for p = 2, while the right most marker is p = 10.

We can see that for an early process shift at 10 observations, the relation seems to break down for larger group sizes. The POD is actually higher for a high value of $\mathcal{L}$. It should also be noted that in this case the group size is very close to the number of observations (9 sensors and 10 observations) which could also cause this to happen. In general, the relation between $\mathcal{L}$ and POD seems to hold up well for different values of τ. For large values of τ, the relation is more pronounced.

5. CONCLUSIONS

This research proposed an online algorithm for finding similar groups of sensors without expert and historical knowledge. An optimization problem was set up using a loss function based on the peak of the posterior prediction distribution. We showed that a relation exists between this loss function and the probability of detection (POD). The simulated annealing (SA) algorithm was used to minimize the loss function, as it found the lowest loss solutions after 1 second of run time. We used the proposed procedure for finding groups of sensors together with the SSMEWMA control chart and therefore termed our procedure the P-SSMEWMA.

The P-SSMEWMA procedure can be applied to specialized industrial processes; such as aerospace, shipping and semiconductors. For verification and comparison, we applied four control charts including the P-SSMEWMA to a case study of a previous generation multibeam wafer scanner. The found groups with the SA algorithm resemble process knowledge and estimated covariance. The simulation experiment based on the case study found that the P-SSMEWMA POD was higher for most combinations of δ and τ. Specifically, the P-SSMEWMA was faster in detecting shifts and almost always outperformed the other control charts when δ was small. Overall, the procedure showed improvements in the POD_2, POD_5 and POD_{10}. A sensitivity analysis revealed that the used loss function is related to the POD of SSMEWMA charts and can be used for a variety of values for δ and τ.

These findings highlight the usefulness of implementing the P-SSMEWMA procedure for high-dimensional, short-run manufacturing processes. A significant increase in the probability of detecting process shifts can easily be obtained by monitoring groups of sensors found by maximizing the posterior peak of the predicted observations. The procedure does not require an extensive investigation of the process. Implementing the P-SSMEWMA has the potential to improve quality control, decreased costs and improved process knowledge.

6. RECOMMENDATIONS

In this section, recommendations are given based on the results of this research. The section is divided into two subsections due to the results being of interest for both industrial applications and future research.

6.1 For industrial implementation

Implementing SPM methods in an industrial setting comes with various trade-offs. One downside of the use of the MLE estimate is that it is not regularized. For practical implementation we would recommend to use a regularized estimate of the covariance, or use a Bayesian approach with priors. If a Bayesian approach is chosen, we recommend to set a prior such that process knowledge is also incorporated.

If a Bayesian approach is not used, we recommend to use the minimum covariance determinant (MCD) as it is less sensitive to outliers, avoiding an estimate of the covariance matrix that is contaminated by outliers. As an alternative, an extra check could be added that removes outliers from the data used for estimation, if for instance the control limit was exceeded.

Finally, we recommend to use the P-SSMEWMA chart in combination with single Q-charts monitoring all sensors. The P-SSMEWMA procedure is relatively complex to implement and does not directly show the sensor measurements. In practice, being able to see the actual measured values in a control chart is beneficial. For example, operators could immediately see which sensors was responsible for

an OOC event. By implementing both charts, OOC events would be detected quickly, and the Q-charts could provide insights about the individual measurements.

6.2 For future research

Future research could look into the viability of combining the partitioning algorithm with other multivariate control charts, such as the HC chart. The SSMEWMA cannot detect changes when p is greater then n, if for instance the HC chart is used, this dependence would not exist and performance could be improved. As an alternative, a constraint could be added to force the group size to be less then n at each time step. This is done mainly to avoid the missed detection of shifts due to too large group sizes.

The HC-chart used in the simulation was applied to the full data set of 96 sensors, reducing the number of sensors could significantly improve the performance of the HC-chart as well. Furthermore, the initial implementation of the HC chart suffered from a slight error in the ARL_{IC} setting. It is recommended for future researcher to carefully review the HC chart implementation.

If a simulation study is performed, it is recommended to perform Monte Carlo simulations with high-performance hardware. This could drastically reduce computational times, consequently improving accuracy of the simulations and control limit settings. Another topic to investigate is how well the partitioning algorithm functions when the normality assumption does not hold. A multivariate Gaussian distribution was assumed for the case study. In practice this assumption might not hold and have an effect on the performance of the control chart.

The loss function used in this research is not normalized by the number of sensors. By taking the natural log of the loss function, we obtain a term that only depends on p. This term always results in a constant when summing over all models, because we have a fixed number of sensors (96 for the case study). Therefore, for future research it is recommended to normalize the loss function by dividing by the number of sensors. Which gives a sort of per-sensor probability at the MAP estimate.

The recursive residuals are produced based on the regression of a sensors left neighbors and upper predecessors. Meaning that ordering matters in the calculation. The effect of different orderings could potentially impact the POD. It would be interesting to closely investigate this effect. Related to this, is the possible extension to a full Bayesian framework for the grouping procedure. Potentially including a proper implementation using the model evidence, priors and sequential updates of all model parameters. The current implementation provides an approximation.

In general, a more extensive sensitivity analysis could be performed. For instance, varying λ and investigating the results could be valuable. Other loss function could also be further investigated, for instance the small sample model selection criteria or evidence based functions. Furthermore, looking into the performance of the P-SSMEWMA for different types of shifts, such as: trends, outliers and variance change, could add to the robustness of the procedure. Lastly, the algorithm for generating groups could be investigated for further improvements. For instance, instead of swapping one edge, multiple edges could be swapped simultaneously.

ACKNOWLEDGMENTS

This research project was conducted in the framework of a student internship provided by ASML Delft. The author would like to thank Ingeborg de Pater from TU Delft for supervising this research. We also would like to extend appreciation for Li [17] for helping clarifying questions about the HC method. Furthermore, the Q-branch team has been very helpful during the research project, data was provided by them as well as help with clean room facilities. Specifically, we thank Paul Scheffers and Irina Rod for their help as supervisors. Finally, we thank Joris Bierkens from TU Delft for a fruitful discussion and help with the mathematical method.

APPENDIX A. APPENDIX SECTION

A.1 Bayesian model evidence based groups

For a model $\mathcal{M}$ for data X_p for p sensors with a set of parameters β_p (regression coefficients) the evidence is given by:

$$p(X_p|\mathcal{M}) = \int p(X_p|\beta_p, \mathcal{M}) p(\beta_p|\mathcal{M}) d\beta_p \tag{A.1}$$

Here $p(X_p|\beta_p, \mathcal{M})$ is the likelihood of the data given the parameters and $p(\beta_p|\mathcal{M}$ is the prior, which in our case can be defined as the posterior of the previous time step since we are doing sequential updates. This first prior can be specified in any way.

Often, Equation A.1 is difficult to calculate since it involves the integral over the parameters. A closed-form solution does not often exists and numerical approximations are normally used [29]. However, closed-form solutions do exist for special cases, specifically when we are dealing with a conjugate-Gaussian prior and Gaussian likelihood (see chapter 2.3 of Bishop for more details [23]). First we need to consider what the likelihood and prior mean in our case. Equation A.1 gives the model evidence at observation n. The model evidence ($p(X_p|\mathcal{M})$) can be written as a product of normalization constants, where the normalization constant for the likelihood is given as: $Z_l = (2\pi)^{\frac{nd}{s}}$ [24]:

$$p(X_p|\mathcal{M}) = \frac{Z_n}{Z_{n-1} Z_l} \tag{A.2}$$

Where Z_n and Z_{n-1} are the normalization constants of the posterior and prior respectively. In order to calculate a closed-form solution for $p(X_p|\mathcal{M})$ we should select the

prior to be a conjugate to the likelihood [24]. This prior has the form of a Normal-inverse-wishart or NIW distribution, which is the multivariate expansion of the gamma distribution. It is defined as follows [24]:

$$\begin{aligned} p(\beta_p|\mathcal{M}) &= NIW(\beta_p, \Sigma|m_0, \kappa_0, \nu_0, S_0) \\ &\in N(\beta_p|m_0, \frac{1}{\kappa_0}\Sigma) \times IW(\Sigma|S_0, \nu_0) \end{aligned} \tag{A.3}$$

Where $N(\beta_p|m_0, \frac{1}{\kappa_0}$ is a to be specified Gaussian scaling function with selected prior mean m_0 and covariance Σ (Σ is often selected to be I as a conservative prior) multiplied by a scaling constant κ_0 that represents how strongly we belief our prior on m_0. S_0 is our prior mean for Σ and ν_0 how strongly we believe this prior. Using the NIW distribution as our prior and using Equation A.2, following a derivation by [24], the evidence of the model at observation n is given by:

$$p(X_p|\mathcal{M}) = \frac{1}{(2\pi)^{\frac{d}{2}}}(\frac{\kappa_{n-1}}{\kappa_n})^{\frac{d}{2}} \frac{|S_{n-1}|^{\frac{\nu_{n-1}}{2}}}{|S_n|^{\frac{\nu_n}{2}}} \frac{\Gamma_d(\frac{\nu_n}{2})}{\Gamma_d(\frac{\nu_{n-1}}{2})} \tag{A.4}$$

Where: $\kappa_n = \kappa_0 + n$, $\nu_n = \nu_0 + n$ and $S_n = S_0 + S + \kappa_0 m_0 m_0^T - \kappa_n m_n m_n^T$ with S as the uncentered sum-of-squares matrix (estimate of covariance).

Equation A.4 can be seen as the evidence of model $\mathcal{M}_i$ for a subset X_p of the data D_n at observation n. Interestingly, the equation is a trade-off between the prior and estimate obtained using the data. Taking the natural log of the model evidence (Equation A.4) gives a similar expression to Equation 2.29 but now with a prior term from the previous time step and weight constants κ and ν:

$$\begin{aligned} p(X_p|\mathcal{M}) = &-\frac{d}{2}ln(2\pi) - \frac{\nu_n}{2}ln|S_n| - \frac{d}{2}ln(\kappa_n) + ln(\Gamma_d(\frac{\nu_n}{2})) \\ &+\frac{d}{2}ln(\kappa_{n-1}) + \frac{\nu_{n-1}}{2}ln|S_{n-1}| - ln(\Gamma(\frac{\nu_{n-1}}{2})) \end{aligned} \tag{A.5}$$

Maximizing Equation A.5 gives the model with the highest evidence. The term $-\frac{d}{2}ln(2\pi) - \frac{\nu_n}{2}ln|S_n|$ resembles Equation 2.29. If we perform a batch update to find the model evidence, $ln|S_{n-1}| = ln|S_0| = I$ results in Equation A.5 mainly being a function of $-\frac{d}{2}ln(2\pi) - \frac{\nu_n}{2}ln|S_n|$ and weights that determine how much we trust the data or the prior.

A.2 Control chart verification

The implemented control charts from literature were verified to ensure correct implementation. The results of this verification step are shown below.

A.2.1 SSMEWMA verification

To verify the implementation of the SSMEWMA chart, table 1 was used for reference. The values for ARL were taken from the paper [7] and the corresponding h was calculated with the implemented code. The values found with the simulation code closely match the values found in the paper, see table A.2.2

Table 3. Simulation verification results rounded to two decimals with SD of statistic as error. 100 runs were performed. For group size 2.

λ	Paper h	Simulation code h
0.05	6.242	6.26 ± 0.051
0.10	7.262	7.30 ± 0.11
0.15	7.832	7.64 ± 0.16
0.20	8.211	8.19 ± 0.22
0.25	8.458	8.38 ± 0.29

Table 4. Simulation verification results rounded to two decimals with SE as error for HC chart from Li et al. In total, 50 runs were performed per ARL for a run length of 40 observations.

δ	Paper ARL	Simulation code ARL
0.5	9.6	13.0 ± 6.04
1.0	6.6	6.55 ± 3.03
1.5	3.7	3.12 ± 0.31
2.0	3	2.96 ± 0.16
2.5	3	3.0 ± 0.0
3.0	3	3.0 ± 0.0

Furthermore, table 3 in [7] gives an example with process values and the corresponding R, U and M matrices. The values found for this example were also compared with the results from the table and were found to match.

A.2.2 HC verification

For small shifts the ARL_{OOC} is large. To accurately asses the ARL_{OOC}, the process should run indefinitely until the limit is exceeded. However, in practice this was unfeasible due to long computation times. The cut-off was placed at 15 observations after the shift. For this reason, the ARL_{OOC} for small shifts is lower then expected, due to the very long ARL_{OOC} not being counted. The standard error is consequently also larger, as fewer samples could be taken. Therefore, verification was only done for the top right results of table 1, as the ARL is relatively low and the total RL after the shift relatively high.

From Table A.2.2 it can be seen that the simulated values match closely with the original papers results, within standard error bounds.

A.3 Control limit setting

Table 5. Numerically found control limits. For the HC chart, we assume a constant limit.

Limit	Value
α(Q-chart)	0.0006446
h (R-SSMEWMA)	12.576804
correction (P-SSMEWMA)	0.952502
$\bar{h}(HC-chart)$	2.20

For the full table of values found for the P-SSMEWMA we refer to the Git hub code used for performing the simulations.

REFERENCES

[1] Moscato, D. R. (2006). Mastering Statistical Process Control: A Handbook for Performance Improvement Using Cases. *Benchmarking: An International Journal* **13**(1/2) 259–261. https://doi.org/10.1108/14635770610709086.

[2] van Stijn, M. C. A. Change Detection in System Parameters of Lithography Machines (2018). Master's thesis, Eindhoven University of Technology. Volume 7, Page 2, ISSN 2277–5668.

[3] Dahari, S., Talib, M. A. and Ghapor, A. A. (2024). Robust Control Chart Application in Semiconductor Manufacturing Process. *Journal of Advanced Research in Applied Sciences and Engineering Technology* **43**(2) 203–219. https://doi.org/10.37934/araset.43.2.203219.

[4] Jones, L. A., Champ, C. W. and Rigdon, S. E. (2001). The Performance of Exponentially Weighted Moving Average Charts with Estimated Parameters. *Technometrics* **43**(2) 156–167.

[5] Roes, C. B. SPC bij laag volume en kleine series fabricage (1996). PhD thesis, Universiteit van Amsterdam.

[6] International Organization for Standardization (2017). Control Charts — Part 8: Charting Techniques for Short Runs and Small Mixed Batches. Standard, INTERNATIONAL STANDARD, Geneva, CH.

[7] Hawkins, D. M. and Maboudou-Tchao, E. M. (2007). Self-Starting Multivariate Exponentially Weighted Moving Average Control Charting. *Technometrics* **49**(2) 199–209. https://doi.org/10.1198/004017007000000083. https://doi.org/10.1198/004017007000000083.

[8] Chen, S. X. and Qin, Y. -L. (2010). A Two-Sample Test for High-Dimensional Data with Applications to Gene-Set Testing. *The Annals of Statistics* **38**(2) 808–835. https://doi.org/10.1214/09-AOS716.

[9] Wiederhold, M., Greipel, J., Ottone, R. and Schmitt, R. (2016). Clustering of Similar Processes for the Application of Statistical Process Control in Small Batch and Job Production. *International Journal of Metrology and Quality Engineering* **7**(4).

[10] Greipel, J. S., Nottenkämper, G. and Schmitt, R. H. (2020). Comparison of Grouping Algorithms to Increase the Sample Size for Statistical Process Control. *SN Applied Sciences* **2**(5) 935. https://doi.org/10.1007/s42452-020-2728-x.

[11] BSI British Standards (2008). Guide to Statistical Process Control (SPC) Charts for Variables – Part 3: Charting Techniques for Short Runs and Small Mixed Batches. Standard, BSI British Standards, Milton Keynes, GB.

[12] Quesenberry, C. P. (1991). SPC Q Charts for Start-Up Processes and Short or Long Runs. *Journal of Quality Technology* **23**(3) 213–224. https://doi.org/10.1080/00224065.1991.11979327. https://doi.org/10.1080/00224065.1991.11979327.

[13] ASML (2025). *HMI eScan 1100*. Accessed: 2025-04-08. https://www.asml.com/en/products/metrology-and-inspection-systems/hmi-escan-1100.

[14] Quesenberry, C. P. (1995). On Properties of Q Charts for Variables. *Journal of Quality Technology* **27**(3) 184–203.

[15] Laurijsse, R. Self-starting Process Monitoring for High-tech Manufacturing (2021). PhD thesis, TU Eindhoven.

[16] Leung, M. S. H. and Corcoran, J. (2021). Evaluating the Probability of Detection Capability of Permanently Installed Sensors Using a Structural Integrity Informed Approach. *Journal of Nondestructive Evaluation* **40**(3) 82. https://doi.org/10.1007/s10921-021-00806-5.

[17] Li, Y., Liu, Y., Zou, C. and Jiang, W. (2014). A Self-Starting Control Chart for High-Dimensional Short-Run Processes. *International Journal of Production Research* **52**(2) 445–461. https://doi.org/10.1080/00207543.2013.832001. https://doi.org/10.1080/00207543.2013.832001.

[18] Hawkins, D. M. (1987). Self-Starting Cusum Charts for Location and Scale. *Journal of the Royal Statistical Society. Series D (The Statistician)* **36**(4) 299–316.

[19] Hawkins, D. M. (1991). Diagnostics for Use With Regression Recursive Residuals. *Technometrics* **33**(2) 221–234. https://doi.org/10.1080/00401706.1991.10484809. https://www.tandfonline.com/doi/pdf/10.1080/00401706.1991.10484809.

[20] Chen, Z. (2003). Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. *Statistics* **182**. https://doi.org/10.1080/02331880309257.

[21] Kianifard, F. and Swallow, W. H. (1996). A Review of the Development and Application of Recursive Residuals in Linear Models. *Journal of the American Statistical Association* **91**(433) 391–400. Accessed 2025-06-09.

[22] Galpin, J. S. and Hawkins, D. M. (1984). The Use of Recursive Residuals in Checking Model Fit in Linear Regression. *The American Statistician* **38**(2) 94–105. Accessed 2025-06-08.

[23] Bishop, C. M. and Nasrabadi, N. M. (2006) *Pattern Recognition and Machine Learning* **4**. Springer.

[24] Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning series.* MIT Press. https://books.google.nl/books?id=RC43AgAAQBAJ.

[25] Hannart, A. and Naveau, P. (2014). Estimating High Dimensional Covariance Matrices: A New Look at the Gaussian Conjugate Framework. *Journal of Multivariate Analysis* **131** 149–162. https://doi.org/10.1016/j.jmva.2014.06.001.

[26] Tian, C. and Kalia, C. (2023). *Stirling Numbers of the Second Kind*. MIT PRIMES Circle, Spring 2023.

[27] Florek, K., Łukasiewicz, J., Perkal, J., Steinhaus, H. and Zubrzycki, S. (1951). Taksonomia wrocławska. *Przegląd Antropologiczny* **17**(2) 193–210.

[28] Brent, R. P. (1971). An Algorithm with Guaranteed Convergence for Finding a Zero of a Function. *The Computer Journal* **14**(4) 422–425.

[29] Särkkä, S. (2013) *Bayesian Filtering and Smoothing. Bayesian Filtering and Smoothing*. Cambridge University Press. https://books.google.nl/books?id=5VIsAAAAQBAJ.

Thijs Beene. Delft, Delft University of technology, the Netherlands.