NTNU
Norwegian University of
Science and Technology

Methods for fault detection in an artificial pancreas

Karl Arthur Frelsøy Unstad

Master of Science in Cybernetics and Robotics
Submission date: February 2018
Supervisor: Øyvind Stavdahl, ITK
Co-supervisor: Anders Fougner, ITK
Konstanze Kölle, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

MSc thesis spring 2017

Title: Methods for fault detection in an artificial pancreas

Background: The Artificial Pancreas Trondheim (APT) research group is working towards an artificial pancreas (AP) for people with diabetes mellitus type 1 (DM1). An AP is a fully automated system that adjusts the exogenous insulin administration based on continuous glucose monitoring (CGM) and possibly other measured variables. The main components of an AP are a glucose sensor, an insulin infusion pump including an insulin reservoir and a delivering tube, and a control unit computing the needed amount of insulin.

Faults can occur in all of these components and impair the safety of the user if the system behaves abnormally. The control unit should therefore include functions for fault detection and diagnosis that trigger an adjusted operation upon fault detection.

A mathematical model and simulator describing the glucose-insulin metabolism for meal scenarios in DM1 has been approved by the American Food and Drug Administration (FDA) for preclinical trials. This simulator, possibly also a clinical data set (currently being collected), shall be used in this thesis as basis to study the potential of pattern recognition for fault detection and isolation based on CGM.

Tasks:

1) Give an overview of methods that have been used for fault detection in artificial pancreas. Give special attention to methods used on more than one type of faults (i.e. >2 classes). Which methods have not been used?

2) Based on the literature review, select methods for fault detection in artificial pancreas.

3) Implement the selected methods.

4) Train/test/review the methods on the simulator. If time permits, test the methods also on clinical data (if the collected data set is suitable).

Supervisor: Øyvind Stavdahl
Co-supervisors: Konstanze Kölle, Anders Fougner

Abstract

This thesis is concerned with research related to diabetes treatment, and more specifically with treatment methods via artificial pancreas. A fault detection system is developed and implemented in the APT-simulator. The presented system is classification based, and utilizes SPE monitoring charts. Both a personal and global models are achieved. It is intended that the system is capable of detecting several types of faults, and the presented system accomplishes this by detecting up to 4 different types of faults. In addition to being capable of detecting the different types of faults, the presented system is also able to differentiate between the faults. The fault detection method is tested in the APT-simulator, and is capable of detecting the following faults: loss of amplitude, pressure induced sensor attenuation (PISA), infusion faults, and jumping signal. Furthermore, the method has an acceptable low number of false positives.

Sammendrag

Denne avhandlingen omhandler forsking relatert til behandling av diabetes, og mer spesifikt, sees det på behandling av diabetes ved bruk av kuntig pancreas. En feildeteksjonsmetode er utviklet og implentert i APT-simulatoren. Systemet som presenteres, er basert på klassifisering og tar i bruk det som på engelsk kalles SPE-monitoring charts. Både personlige og globale modeller oppnås. Det er hensikten at systemet er i stand til å detektere flere typer feil, og systemet som presenteres, innfrir dette ved å detektere opp til 4 forskjellige typer feil. I tillegg til å være i stand til å detektere forskjellige typer feil, klarer systemet også å differensiere mellom disse feilene. Feildeteksjonsmetoden er testet i APT-simulatoren og er i stand til å detektere følgende typer feil: tap av amplitude, trykkindusert sensor demping (PISA), infusjonsfeil og alternerende signal. Dessuten har metoden et akseptabelt lavt antall falske feildeteksjoner.

Preface

This thesis is submitted as part of the Master of Science degree at the Norwegian University of Science and Technology, Department of Engineering Cybernetics. The work presented has been carried out in collaboration with the Artificial Pancreas Trondheim (APT) research group.

There are many people I would like to thank for their help, and I want to start by thanking my supervisor Øyvind Savdahl for introducing me to my co-supervisors Konstanze Kölle and Anders Fougner. Thank you Konstanze and Anders for sharing your time and knowledge with me, and for your support and encouragement. I am also truly thankful for the opportunity to work with Artificial Pancreas Trondheim. It has been an incredibly inspiring experience.

I want to thank close my friend Tai for valuable discussions and support, and for all the hours spent together at Gløshaugen. I am grateful for all the help you have given me over the years. I also want to thank my family for the motivation and encouragement you have always given me.

Last but not least, I want to thank my fiancée, the love of my life, Renate. This thesis has been a long and challenging experience, and your unwavering support and motivation has meant the world to me. I am truly blessed to have you by my side.

Karl Arthur Unstad,
Trondheim, February 2018

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Diabetes mellitus is a chronic metabolic disorder where either the pancreas stops producing insulin, or the cells do not respond to the produced insulin. There are three main types of diabetes: Type 1, where the pancreas fails to produce insulin. Type 2, which starts with insulin resistance and a lack of insulin may develop as the disease develops. The third form is Gestational diabetes occurs in pregnant women.

The number of people with dieabetes is rising, and according to the [?], 425 million adults have diabetes. Out of those 1 in 2 are undiagnosed. This means that 1 out of 11 adults in the world can not properly regulate their blood glucose levels. As a result, they live with reduced life quality, and have higher risk of disabling and life-threatening health problems.

Good control of blood glucose levels will reduce the risk of complications and there are several potential treatments. A healthy diet, exercise, weight loss, and use of insulin combined with normal cholesterol and blood pressure levels will greatly reduce the risks.

The "holy grail" of diabetes treatment is the artificial pancreas (AP), a closed-loop control system that regulates the user's blood glucose level by infusing insulin[12]. Advances in technology and research have been done and today we are closer to an AP than ever before.

With an AP, as with any other system, there might be faults and because of the seriousness of the disorder they might be dangerous for the user. Effective fault detection is therefore an essential part of an AP, and necessary to prevent both short and long term complications of diabetes.

1.1 Related Work

The greatest contribution to this field is the Type 1 Diabetes Metabolic Simulator (T1DMS) from the UVA/Padova research group. The group is a collaboration between the University of Virginia (USA) and the University of Padova (Italy), and in 2008 the T1DMS was approved as a substitute for preclinical trials by the U.S. Food and Drug Administration (FDA). As a result *in silico* experiments was made possible, and the AP research was accelerated. T1DM and improved sensor technology has made the idea of the AP possible, and consequently the fault detection related research has increased the last decade.

[7] proposed a continuous-discrete extended Kalman filter for detection of spikes and drift. The method is tested by simulations done in the Medtronic virtual patient model.

In [13] subject-specific recursive linear time series models was introduced to predict future glucose concentrations for prevention of hypoglycaemia.

Multiway principal component analysis (PCA) was used in [15] to detect different faults. Further development was done in [14] where the subjects was provided with a sport armband which gave the fault detection additional information.

In [4] CGM and continuous subcutaneous insulin infusion (CSII) was combined with a black-box model to detect spikes, loss of sensitivity in the sensor and failure of the pump to deliver insulin. The method was tested during night-time with the UVA/Padova-simulator. In [3] the method was extended to the whole day, including meals, and meal information was used to further bolster the fault detection.

[1] developed algorithms to detect infusion set failures and sensor signal anomalies, and both in-patient and in-patient studies are presented. The detection of Pressure induce sensor attenuation is later improved upon in [2]. Some years later Baysal was involved in [5] where continuous glucose monitoring (CGM) was used to detect losses in infusion set actuation.

[17, 18] uses Principal Component Analysis (PCA) and monitoring charts to detect spikes and transient loss of sensitivity of the sensor. At a late stage the work was further developed in [11, 10], where a classification based fault detection method based on PCA was made to detect spikes and loss of sensitivity. A global model based on training data independent from the user is also proposed. Zhao and Songs methods for fault detection was tested in the UVA/Padova-simulator.

1.2 Scope and emphasis

The aim of the thesis is to further this important research, and to be a part of the development towards a fully functional autonomous AP. An AP traditionally consisting of one sensor estimating blood glucose concentration, a controller, and a pump infusing insulin. Ideally an AP should consist of as few parts as possible, therefore the thesis will be based on the assumption that only CGM-signals and insulin infusion rate (IIR) are available for the fault detection. The emphasis is put on development of a fault detection method that is capable of detecting more than one type of fault.

The work done by [10] is based solely on the information provided by the CGM, and is not dependent other information such as meals. The method is also able to make a global fault detection model, and have the potential to detect several types of faults. It is therefore used as a starting point for this thesis. Due to lack of documentation a similar, but partly different fault detection method is developed and explained in Chapter 3.

The developed method is tested on an extended scenario consisting of three meals and 5 faults. This is done by implementing the fault detection in the APT-simulator, a diabetes simulator provided by the research group Artificial Pancreas Trondheim (APT).

1.3 Outline of the thesis

Chapter 2 gives an introduction to the theory used the thesis. Chapter 3 describes the fault detection method step-by-step and presents the different alternatives it has to offer. Chapter 4 presents the results and other observations from simulations of the fault detection method using the APT-simulator. Chapter 4.5 gives a short summary of what is accomplished and the conclusion of the thesis, before Chapter 5 presents suggestions for potential improvements and ideas for future work.

Chapter 2

Background Theory

2.1 Principal Component Analysis

2.1.1 Introduction

Principal component analysis (PCA) aims to reduce the dimensionality of data sets consisting of large numbers of interrelated variables, while at the same time retaining as much as possible of the variation present in the data set. This is done by transforming the data set to a new set of uncorrelated variables, the principal components (PCs), which are ordered so that the first few retain most of the variation present in all of the original variables [6]. The data set $\mathbf{X}$ is organized with distinct observations along each row and the columns are recorded values for each observation. The data set is then decomposed into three matrices.

$$\mathbf{X} = \mathbf{Z}\mathbf{P}^T + \mathbf{E}$$

where $\mathbf{Z}\mathbf{P}^T$ is the bilinear model of the data set and $\mathbf{E}$ is the residual matrix. $\mathbf{Z}$ is the score matrix containing the transformed coordinates of the observations in $\mathbf{X}$, and $\mathbf{P}$ is the loading matrix which contains the axis/directions of the transformed coordinate system.

2.1.2 Mathematical description

Suppose the data matrix $\mathbf{X}$ consists of vectors $\mathbf{x}_i$ with m random variables on the form $\mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^m)$. We want to examine the variance of the m variables and the covariance or correlations between the variables. Unless m is small it will not be very helpful to look at the m variances and correlations or covariances. A better approach is to find a few variables that preserve most of the information in the variances and correlations or covariances. PCA does not ignore covariances and correlations, but the main focus is on variance. First we search for a linear function $\mathbf{p}_1^T\mathbf{x}_i$ of the elements in $\mathbf{x}_i$ for which we want to maximize the variance. $\mathbf{p_1}$ is a vector of m constants $p_{11}, p_{12}, \ldots, p_{1m}$ so that

$$\begin{aligned}\mathbf{p}_1^T\mathbf{x_i} &= p_{11}x_i^1 + p_{12}x_i^2 + \cdots + p_{1m}x_i^p \\ &= \sum_{j=1}^{m} p_{1j}x_i^j\end{aligned}$$

The next step is to find a linear function $\mathbf{p}_2^T\mathbf{x}_i$ uncorrelated with $\mathbf{p}_1^T\mathbf{x_i}$ having maximum variance. As we progress $\mathbf{p}_k^T\mathbf{x_i}$ is found with maximum variance and uncorrelated with $\mathbf{p}_1^T\mathbf{x_i}, \mathbf{p}_2^T\mathbf{x_i}, \ldots, \mathbf{p}_{k-1}^T\mathbf{x_i}$. The variable $\mathbf{p}_k^T\mathbf{x_i}$ is the kth PC and up to m PCs could be found, but generally it hoped that most of the variation in x_i will be accounted for by $d \ll m$ PCs. Figure 2.1 shows a plot

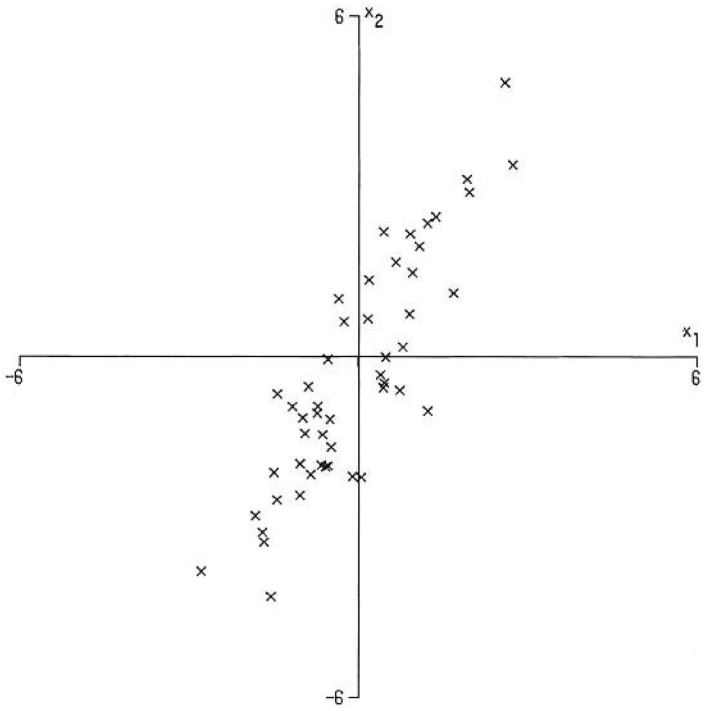

Figure 2.1: Plot of 50 observations on two variables x_1 and x_2. [6]

of 50 observations with the two highly correlated variables x_1 and x_2 (not to be confused with $\mathbf{x}_1$ and $\mathbf{x}_2$). If we transform the data to the PCs z_1 and z_2 the resulting plot can be seen in Figure 2.2. We can now see that there are a greater variation in the z_1-direction than in both the original variables. On the other hand there is very little variation in the direction of z_2. In the general case, if a data set has $m > 2$ variables with substantial correlation between them, the first PCs will represent most of the variation of the original variables. The last PCs will identify directions with little variation, the directions with near-constant linear relationship among the original variables.

For the derivation of the PCs consider $\mathbf{p}_1^T\mathbf{x}_i$, where the vector $\mathbf{p}_1$ maximizes $\text{var}\left(\mathbf{p}_1^T\mathbf{x}_i\right) = \mathbf{p}_1^T\mathbf{\Sigma}\mathbf{p}_1$, where $\mathbf{\Sigma}$ is the covariance matrix. The maximum can

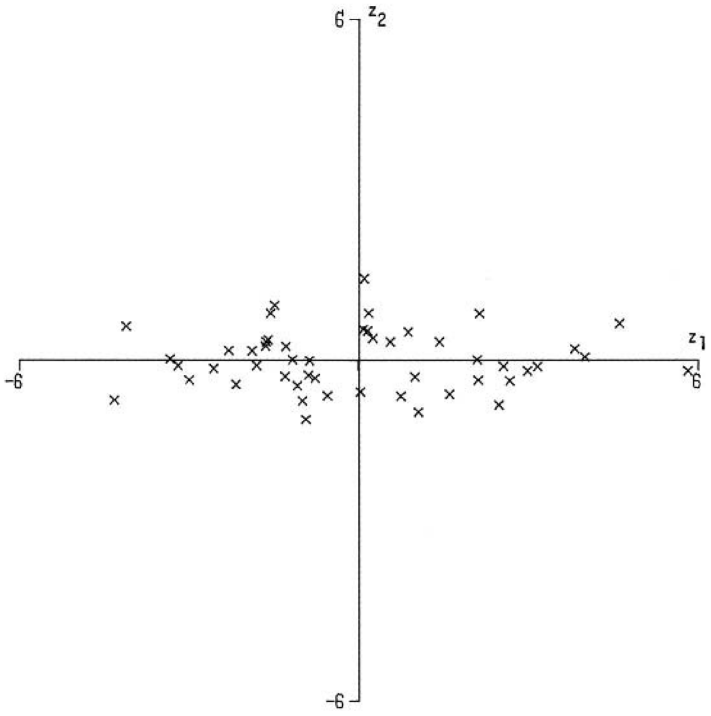

Figure 2.2: Plot of the 50 observations from Figure 2.1 with respect to their PCs z_1 and z_2. [6]

not be reached with a finite $\mathbf{p}_1$ so a normalization constraint is imposed. The constraint used is $\mathbf{p}_1^T\mathbf{p}_1 = 1$. To maximize $\mathbf{p}_1^T\mathbf{\Sigma}\mathbf{p}_1$ with regards to $\mathbf{p}_1^T\mathbf{p}_1 = 1$ Lagrange multipliers is used. Maximize

$$\mathbf{p}_1^T\mathbf{\Sigma}\mathbf{p}_1 - \lambda\left(\mathbf{p}_1^T\mathbf{p}_1 - 1\right),$$

where λ is a Lagrange multiplier. Differentiation with respect to $\mathbf{p}_1$ then gives

$$\left(\mathbf{\Sigma} - \lambda\mathbf{I}_m\right)\mathbf{p}_1 = 0,$$

where $\mathbf{I}_m$ is the identity matrix with dimensions $(m \times m)$. This results in λ being an eigenvalue of $\mathbf{\Sigma}$ with $\mathbf{p}_1$ as the corresponding eigenvector. To decide which of the m eigenvector maximizes the variance of $\mathbf{p}_1^T\mathbf{x}_i$ we need to maximize

$$\mathbf{p}_1^T\mathbf{\Sigma}\mathbf{p}_1 = \mathbf{p}_1^T\lambda\mathbf{p}_1 = \lambda\mathbf{p}_1^T\mathbf{p}_1 = \lambda.$$

In other words we need λ as large as possible. Therefore, $\mathbf{p}_1$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{\Sigma}$, and $\text{var}\left(\mathbf{p}_1^T\mathbf{x}_i\right) = \mathbf{p}_1^T\mathbf{\Sigma}\mathbf{p}_1 = \lambda_1$, the largest eigenvalue.

In general, the kth PC of $\mathbf{x}_i$ is $\mathbf{p}_k^T\mathbf{x}_i$ and $\text{var}\left(\mathbf{p}_k^T\mathbf{x}_i\right) = \lambda_k$, where λ_k is the kth largest eigenvalue of $\mathbf{\Sigma}$ and $\mathbf{p}_k$ is the corresponding eigenvector. Proof for $k > 1$ can be found in **??**, but is not needed to understand this thesis. We then get the model on the form

$$\mathbf{Z} = \mathbf{XP}, \tag{2.1}$$

where $\mathbf{Z}$ in on the form $z_{ij} = \mathbf{p}_i \mathbf{x}_j$. This leads us back to the expression given in 2.1.1:

$$\mathbf{X} = \mathbf{ZP}^T + \mathbf{E} \tag{2.2}$$

where $\mathbf{E}$, the residual space, represents the information not used in the model, and depends on the number of PCs used. The fewer PCs used the more information is discarded into the residual matrix section $\mathbf{E}$.

2.1.3 Monitoring charts

As described by [17] PCA-models can be used to develop PCA based Monitoring Chart. Squared prediction error (SPE), also known as Q-statistic, is a lack-of-fit statistic calculated as the sum of squares for each row of E. Equations 2.1 and 2.2 gives us

$$\mathbf{E} = \mathbf{X} - \mathbf{XPP}^T.$$

SPE for observations i is then given by

$$\mathrm{SPE}_i = \mathbf{e}_i \mathbf{e}_i^t$$

where $\mathbf{e}_i$ is the ith row of $\mathbf{E}$. The total SPE-vector is given by

$$\mathrm{SPE} = \mathrm{diag}\left(\mathbf{E}^2\right)$$

where $\mathrm{diag}\left(\mathbf{E}^2\right)$ represents the diagonal of $\mathbf{E}^2$. When a new observation $\mathbf{x}_{\mathrm{new}}$ is available it is preprocessed using the same normalization information as was used for $\mathbf{X}$. Then the PCA-model is used for calculating the new SPE-value

$$\mathrm{SPE}_{\mathrm{new}} = \mathbf{e}_{\mathrm{new}} \mathbf{e}_{\mathrm{new}}^T,$$

where $\mathbf{e}_{\mathrm{new}} = \mathbf{x}_{\mathrm{new}} - \mathbf{x}_{\mathrm{new}} \mathbf{PP}^T$. The new SPE-value is then compared to a predefined control limit. If it is large and go out of the predefined normal region, it is not representative of the process modelled and something is wrong. If not the new observation is normal and expected.

2.2 Preprocessing

Before modelling the data set it is benificial to perform preprocessing [16]. Preprocessing the data greatly affects the outcome of the data analysis, and [16] concluded that autoscaling and range scaling performed better than the other pretreatment methods. As autoscaling is the most fitting preprocessing method it will be used in this thesis. Autoscaling consists of two parts, mean-centering and normalization to unit variance.

Centering

Centering the data set $\mathbf{X}$ removes the offset from data. It consists of subtracting the column mean from each column in $\mathbf{X}$. The mean is estimated as:

$$\bar{\mathbf{x}}_i = \frac{1}{J}\sum_{i=1}^{J}\mathbf{x}_{ij}$$

The mean-centered data set is then given by:

$$\tilde{\mathbf{x}} = \mathbf{x}_{i,j} - \bar{\mathbf{x}}_i$$

Unit variance

Normalizing the data to unit variance ensures that variables with large magnitudes do not dominate the model, and all variables become equally important. This is done by using the standard deviation as a scaling factor. The standard deviation is estimated as

$$s_i = \sqrt{\frac{\sum_{i=1}^{J}\left(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i\right)^2}{J-1}}$$

The data with unit variance is then given by:

$$\tilde{\mathbf{x}}_{ij} = \frac{\mathbf{x}_{ij}}{s_i}$$

Autoscaling

In combination centering and normalization to unit variance becomes autoscaling and the precessed data is given by

$$\tilde{\mathbf{x}}_{ij} = \frac{\mathbf{x}_{i,j} - \bar{\mathbf{x}}_i}{s_i}$$

2.3 Discriminant analysis classifier

Matlab function `fitcdiscr` makes a classifier based on a table with observations and accompanying table with responses. Matlab function `predict` can then be used to predict class for an unknown observation. `Predict` classifies so as to minimize the expected classification cost:

$$\hat{y} = \arg_{y=1,\ldots,K}\min\sum_{k=1}^{K}\hat{P}\left(k|x\right)C\left(y|k\right)$$

where $\hat{y}$ is the predicted classification, K is the number of classes, $\hat{P}(k|x)$ is the posterior probability of class k for observation x. $C(y|k)$ is the cost of classifying an observation as y when its true class is k. The cost is given by $\text{Cost}(i,j) = 1$ if $i \neq j$, and $\text{Cost}(i,j) = 0$ if $i = j$. An example can be seen in 2.3 where tree different types of flowers is separated by the classifier. This is used in chapter 3, but a more theoretical understanding is not necessary for the implementation.

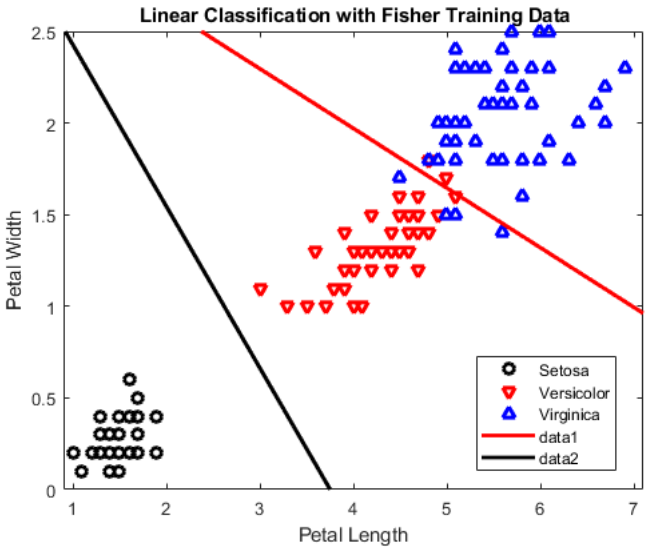

Figure 2.3: Three different types of flowers separated by the classifier. Taken from Matlabs documentation page for `fitcdiscr`.

2.4 Naive Bayesian Classifier

The Naive Bayesian Classifier is based on Bayes' theorem with independence assumptions between the predictors. Despite its simplicity it often outperform more sophisticated classification methods and is therefore widely used. Bayes' theorem makes it possible to calculate the posterior probability $P(c|f)$ from the prior probability of class $P(c)$, prior probability of the predictor $P(f)$ and the probability of a predictor given class $P(f|c)$. It is given by

$$P(c|f) = \frac{P(f|c)\,P(c)}{P(f)}$$

The posterior probability can be calculated by constructing a frequency table for each attribute against the target.The frequency tables are transformed into likelihood tables which can be used as $P(f|c)$. We can then use this to calculate the probability of a class given the predictor, $P(c|f)$.

Chapter 3

Method description and implementation

This section describes the methods used in the thesis. First there is some information about the simulator, the scenario and the faults used. Then the fault detection method is described step-by-step. At the end the fault recognition method is explained.

3.1 The APT simulator

In this thesis a diabetes simulator will be used to test the fault detection method. The simulator used is made by Erlbeck on behalf of the research group Artificial Pancreas Trondheim (APT) and offers different scenarios and different subjects. In total there are 10 adults, 10 adolescents and 10 children, in addition to one average person from each group. In this thesis only the adult group i used. Adult 1-3 are used as test subjects and adult 4-10 are used to generate independent training data for global training. In other words glucose measurements from a group of people used as training data for an independent person. A full description of the simulator can be found in Erlbecks master thesis.

3.1.1 Main scenario

The main scenario used in this thesis consists of 1 day (24 hours) of glucose measurements from a subject. During the 24 hours there are three meals of variable size at semi random times. The meals takes place [1,3], [7,9] and [14,16] hours after simulation start and consists of [35,55], [65,85], [75,95] grams of carbohydrate. Boluses are included, are assumed to be optimal and given at the same time as the meal. The optimal bolus, as described in [8], is given by

$$\text{optimal bolus} = \frac{\text{ingested carbohydrates[g]}}{\text{CR } [\frac{g}{U}]},$$

where CR is the subjects carbohydrate ratio. CR is included for each subject in the simulator. The duration of each meal is set to 15 minutes. To make the scenario more realistic noise is added to the measurements and new measurements are only available every five minutes as that is the standard for sensors today. Figure 3.1 shows the main scenario for the subject Adult 1.

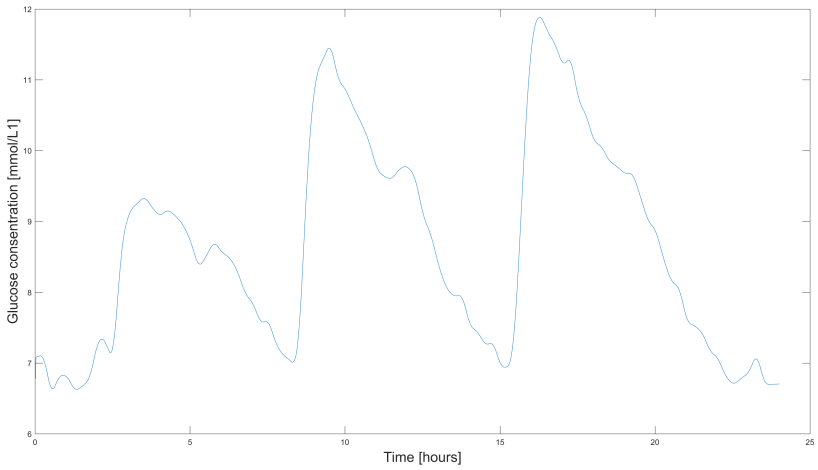

Figure 3.1: Glucose concentration measurements from Adult 1 in the main scenario.

3.1.2 Faults

Then APT-simulator also offers different known faults for an AP. The simulator has three different sensors faults and one infusion fault implemented. In addition spikes was implemented as a part of this thesis. The sensor faults in the simulator are then:

- **Pressure induced sensor attenuation (PISA)**: If the sensor is exposed to pressure, the measured signal drops. This usually happens during nights when the subject is sleeping and is lying on the sensor. The wrong measurement causes a the pump to shut of and the subject is not given the needed insulin. The PISA response has a duration of 15-105 minutes, a depth of 3-5 mmol/L and variable shape.

- **Loss of amplitude**: Towards the end of the sensors lifetime the measurements can be effected by different irregularities. One irregularity that can occur is that the sensor signal gradually loses amplitude.This results in false information to logic unit and the wrong amount of insulin might be given to the subject. The randomized version of loss of amplitude starts after one hour in the main scenario, and the sensor signal is gradually decreased before approaching zero at the end of the day.

- **Jumping signal**: Another irregularity that can occur towards the end of the sensors lifetime is a jumping signal. The sensor signal starts to change between the real measurements and the measurements with an offset. The time of the jumps are randomized and the offset is 1-5 mmol/L.

- **Spike**: When the subjects movement causes movement in the sensor, single measurements that are significantly higher than the ones before and after may occur. A spike consists of a measurement 1-3 mmol/L higher than the real measurement.

The start time of PISA, jumping signal, and spike is random. The infusion fault implemented simulates decreased delivery of insulin from the pump, and is meant to cover a range of different situations that ultimately results in decreased insulin delivery. Examples of such situations are twisted, clogged or loose delivery tube, a faulty pump or infusion set failure. Infusion fault Full description of each fault is found in Unstads term project. An illustration of each fault can be seen in Figure 3.2.

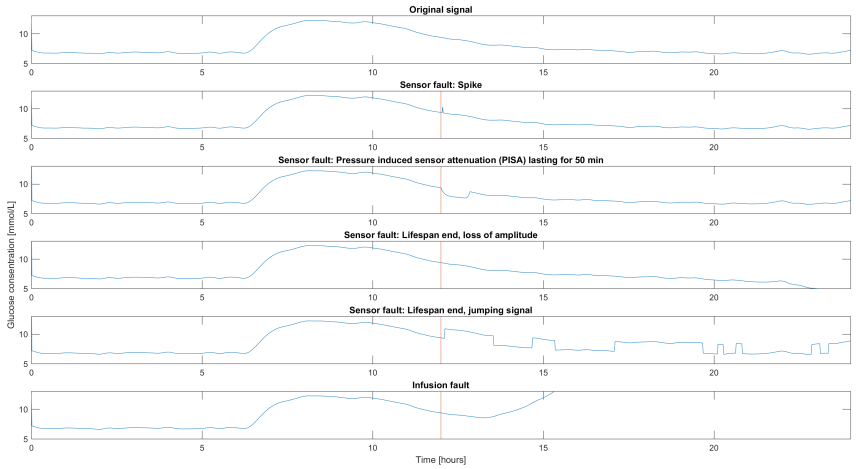

Figure 3.2: Glucose consentration over 24 hours with one meal. Each fault starts after 12 hours.

3.2 Fault detection

This section describes the fault detection method step-by-step. It is inspired by [10] and aims to improve the fault detection method. The target is to monitor changes in glucose characteristics and notify the user when a fault is occurring. Firstly training data is simulated and arranged. The training data is then split into classes depending on different specifications. For each class a control limit is set for the acceptable SPE-values. Each class is then transformed into

feature classes which are used to train a classifier. The classifier is then used on the training data and a probability table for the Naive Bayesian classifier is made. When a new measurement is available the SPE-value for each class is calculated and compared to the control limit. Then the features of the measurement is calculated and the classifier is used to determine the assumed class. The corresponding Bayesian probabilities are then combined with the result from the PCA models from each class. If the final result, which is a probability, is large enough the new measurement is marked as a fault.

3.3 Data arrangement

The training data consists of glucose measurement time series $g(t)$ from the main scenario simulated with the APT-simulator. The time series are broken down into windows of length L and arranged in the data matrix X. A window of length L consists of the L last measurements and is treated as one observation in $\mathbf{X}$. $\mathbf{X}$ is given by:

$$\mathbf{X}(N \times L) = \begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \vdots \\ \mathbf{x}(N) \end{bmatrix}$$

where $\mathbf{x}(k) = [g(k)\ g(k+1)\ \cdots\ g(k+L-1)]$, $N = K - L + 1$ is the number of observations in the data matrix, and K is the number of original time-series observations. $\mathbf{X}$ is then preprocessed by autoscaling.

3.4 Splitting of classes

If one should model the windows in $\mathbf{X}$ it might result in an inaccurate model and some windows might have high SPE-values. By splitting the training data into classes with similar windows the PCA models for each class improves compared to using the . A lot of time and effort went in to replication of the splitting method in [10]. But the article gave to little and to inaccurate information and the split method was not possible to replicate. Therefore two alternative methods are proposed and developed by the author. The first method, Split by SPE, aims to replicate the split method in [10], but in a simplified fashion. The second method, Split by Slope, is an entirely new approach that aims to exploit the linearity of the PCA modelling. An introduction to both methods are given in the following.

3.4.1 Split by SPE

This method aims to replicate the splitting of training data in [10]. For each split the method relies on the range of SPE-values for each class and windows with similar SPE-values are put in classes. In addition other requirements

to variance, linearity and stationarity is set for each class. As a result the training set is split into three classes where class 1 contains roughly twice the number of samples as class 2, and class 2 contains roughly twice the number of samples as class 3.

As an attempt to replicate split method a simple method based on SPE-values is suggested. A PCA model is made for the entire training data. Then the SPE for each window is calculated. The $\lfloor \frac{n \cdot 4}{7} \rfloor$ samples with lowest SPE is placed in class 1, the next $\lfloor \frac{n \cdot 2}{7} \rfloor$ samples is placed in class 2, and the remaining samples is placed in class 3. The resulting three classes are similar to the ones in [11, 10] and give resembling results.

3.4.2 Split by slope

"Linear judgement" and a minimum for variance is two of the criterias in [11, 10] for a class. In an attempt to split the training data into classes that comply with these criteria, without having the possibility to test for linear and stationary judgement, Split by Slope was mate. Split by slope is a simplistic approach that split the training data based on the slope of the window. The slope for $\mathbf{x}(k)$ is given by

$$\mathrm{s} = g(k + L - 1) - g(k)$$

After some initial testing a range given by $\pm 0.0025[\mathrm{mmol/L}] \cdot L$ was found. If the slope of a window is within the range it it put in class 1. If the slope is greater than $0.0025[\mathrm{mmol/L}] \cdot L$ it goes in class 2, and if it less than $-0.0025[\mathrm{mmol/L}] \cdot L$ it goes in class 3. The aim is to class together windows that are roughly linearly which will give lower variance in each class, and each class is also easier modelled by the PCA. An example of this is given in Figure 3.3 where the start and endpoint of windows with length 10 is plotted.

As we can see from Figure 3.3 each class contains windows that are similar to one another. Hopefully this will result in good PCA-models which again will result in a sensitive fault detection.

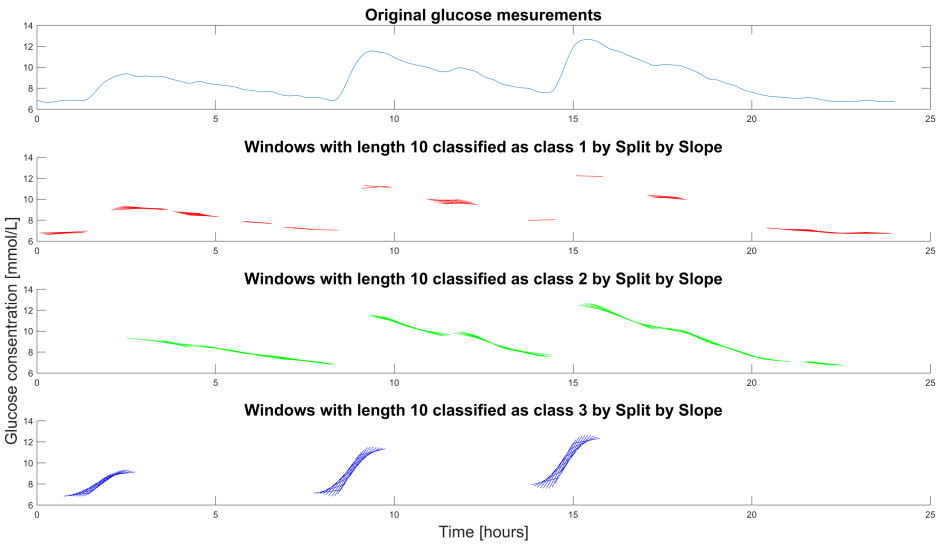

Figure 3.3: Windows of length 10 classified by Split by Slope.

3.5 Control limits

After the training data is split into classes a control limit for acceptable SPE-values is set for each class. Since the training datas is split in a different fashion than in [10], the control limit used might not be suitable. Therefore, three different approaches for deciding the control limits are suggested:

- **Max 1 Limit (M1)**: One PCA-model is made of the entire training data. SPE-values are calculated for each window in each class. The control limit for a class is set to the maximum SPE-value of the windows in that class.

- **Max 2 Limit (M2)**: After the training data is split one PCA-model is made for each class. The control limit for a class is set to the maximum SPE-value of the classes windows in the corresponding PCA-model.

- **Chi-squared-Limit (CS)**: The last limit is the one used in [10], and is given in [9]. The control limit is approximated by a weighted Chi-squared distribution $\text{SPE-limit}_i \approx g_i \chi^2_{h_1,\alpha}$. The weight and the degrees of freedom is given by $g_i = \frac{v_i}{2m_i}$ and $h_i = \frac{2m_i^2}{v_1}$, where v_i and m_i is the variance and the average of all SPE-values in class i.

3.6 Feature selection

To train a classifier to recognise which class a new window belongs to we need to calculate features of the windows. It is important to choose features that

describes the different classes differently so that classification of windows with unknown class is as easy as possible. The precise definition of the features used in [10] is unclear, but these are my interpretations. In addition several more features are added. The features listed in [10] are sliding window slope (SWS) given by

$$\text{SWS} = \frac{\max(\mathbf{x}_i) - \min(\mathbf{x}_i)}{L},$$

first derivative (DG) given by

$$\text{DG} = \frac{g(k+L-1) - g(k)}{L},$$

second derivative (SDG) is calculated by calculating the differences between adjacent elements of the window, and then using the first derivative operation on the resulting vector. Additionally the variance V of the window and the newest glucose measurement $G = g(k+L-1)$ is included. The new features added are window slope (WS)

$$\text{WS} = g(k+L-1) - g(k),$$

the largest difference in the window (LD)

$$\text{LD} = \max(\mathbf{x}_i) - \min(\mathbf{x}_i),$$

the mean M of the window and the number of turns (NT) which is the number of times the window signal changes direction.

The last added feature is the sum of infused insulin (SII) during the window. Insulin infusion rate (IIR) is regarded as unused and significant information, and is believed to possibly contribute to better classification of new windows. The first idea was to include IIR over a longer time period than the window to account for the period the insulin is active in the body. But from a logical and programming standpoint that made some difficulties, as the window can not be constructed before the necessary information is available. For example can a regular window with length 10 be made 50 minutes after the first measurement, as one measurement is available each five minutes. But if infused insulin the last three hours is a feature one have to wait three hours to make the 50 minute window. This seems counter-productive and SII is chosen as a feature of the window. The option to use several windows of different length enables the IIR from different time periods to be accounted for in the system. The resulting feature vector is then given by

$$\text{Features}(\mathbf{x}_i) = [\text{SWS DG SDG V G WS LD M NT SII}]$$

If Split by Slope is used only feature needed is the slope. It is easily calculated from the window, and the window is classified accordingly.

3.7 Classifier

Each class is transformed into feature classes containing the feature vectors of each window in the corresponding class. These feature classes are then used to train a classifier with the Matlab commando `fitcdiscr`. As far as possible only the default linear options are used, but as a result of the inclusion of SII some classes get zero within-class variance. The linear classifier is then unable to use training set. In those instances a pseudo-linear classifier is used instead.

The linear classifier are then used on the feature classes to make a (3×3) frequency table containing the predicted class and the real class. The frequency table is then transformed into a likelihood table where each row gives the probability for each class given the prediction of the linear classifier. The higher values along the diagonal of the likelihood table the better is the classifier, and a perfect classifier the resulting likelihood table would equal the identity matrix. The three values along the diagonal represents the probability that a new window classified as class 1 actually is from class 1, and similar for class 2 and 3. This probability table is used as a basis for the Naive Bayesian classifier.

3.8 Online fault detection

When a new window is available it is scaled with the same mean and standard deviation used to autoscale $\mathbf{X}$. The SPE-value for the new window is then calculated as in **??** for each of the tree classes. The SPE-value is then compared with the corresponding control limits of each class. If the SPE-value is lower than the control limit result_i for class i equals to 0. If it is higher than the control limit result_i is set equal to 1.

The features of the window is then calculated and given to the linear classifier. Based on the classifiers prediction the corresponding row in the probability table made in section 3.7 is extracted and used for $P(C_i|F)$. The total result is then given by

$$\text{result}_{\text{total}} = \sum_{i=1}^{C=3} P(C_i|F) \cdot \text{result}_i,$$

where C is number of classes, C_i is class i, F is the prediction of the linear classifier and $P(C_i|F)$ is the probability for class i given the prediction F. result_{total} is a probability and if it is higher than a chosen fault limit the new window is marked as a fault. The fault limit chosen depends on the split method, type of control limit, and the window length, but generally lies in the interval [0.9 - 1].

3.9 Choice of window lengths

Different faults behave differently and have different manifestations in the glucose measurements. Therefore it follows that it might be beneficial with

different window lengths for different faults. Therefore an extension of the method in [10] is made, and the option to use several windows of different length at the same time is implemented. Each is treated by the same method outlined in this section. If one or more of the windows is marked as a fault, the system as a whole marks the new window as a fault.

3.10 Fault recognition

The method in [10] is only able to detect faults, but is unable to differentiate between the types of faults. This is suboptimal as the course of action after an alarm is dependent on the fault. For example the action taken to a PISA is to remove the pressure from the sensor, but for a infusion fault measures for lowering the glucose concentration needs to be taken. As the origin of the alarm is important an procedure to recognise the fault is proposed. The main scenario is simulated with each fault. When a window is marked as a fault it is saved as an example window of the current fault. This is done for each fault until we have a sufficient number of detected windows. These windows and their known fault is then used to train a classifier using the Matlab commando `fitcdiscr`. We now have a classifier that can predict which fault has occurred when a new window is marked as a fault. If windows of different lengths are marked as faults at the same time, the most predicted fault is chosen. In other words if three windows are marked as faulty, and the three predictions are "PISA", "PISA" and "Spike", "PISA" is chosen as the recognised fault.

Chapter 4

Simulations and results

In this chapter the results of simulations are presented. First is view of some plots from each method. Then the split methods and different limits are tested with different window lengths on the same faults and compared. The best combinations is then used for fault detection and recognitions both in personal training and in global training. In addition the selected methods are tested on slowly decreasing glucose concentration to test sensitivity for low blood sugar.

Features

As said in 3.6 if we Split by Slope the slope can be calculated directly for new windows, and classification is not a problem. For Split by SPE it is a bit more complicated and we need to calculate the features of the new window for the classifier. Some features of a window with length 10 can be seen in Figure 4.1.

First it must be mentioned that the SWS plot in Figure 4.1 is similar to a plot in [10]. In other words the Split by SPE method bears a resemblence with the split method used in [10], although it is far easier. This is a good indication for the fault detection as a whole. As for the DG, SDG and V we can see that none of them gives a direct correlation between the features and the class, and needer does the features not shown. But common for all of them is that they give an indication for which class is likely. For example a new window with high variance is more likely to belong in class 3. If we summarize all these indications it will give a good idea of what class the new window belongs too.

Inclusion of IIR

One of the improvements done compared to the fault detection method in [10] was the inclusion of IIR information. As it turns out it makes a positive difference for the linear classifier. As explained in 3.7 a probability table is made where each row gives the probability of a class given the prediction of the classifier. The goal is to get as high as possible percentages along the diagonal. In Table 4.1 the three diagonal values for 5 different training sets is calculated

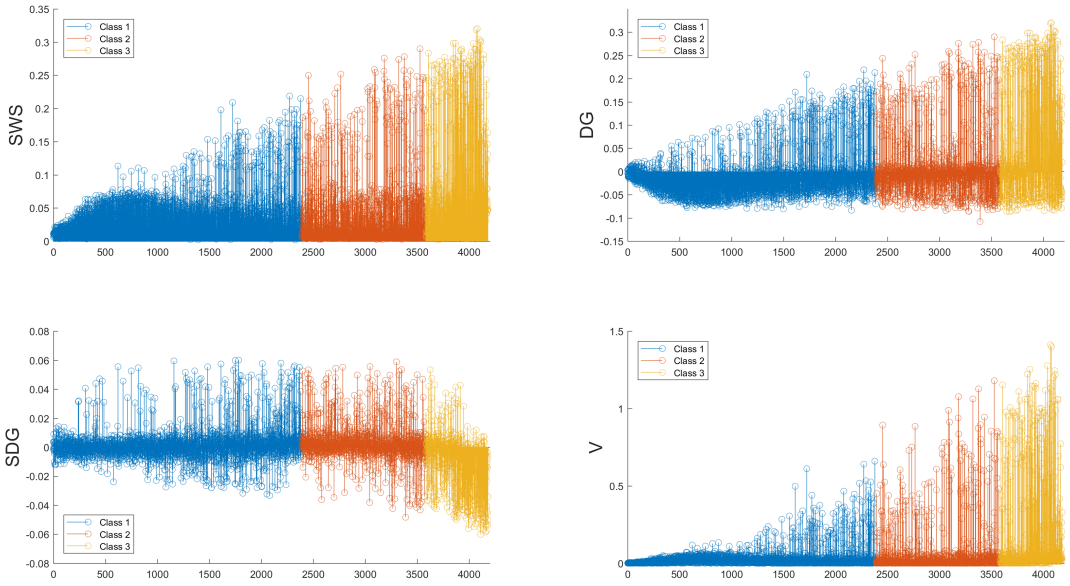

Figure 4.1: Sliding window slope, first derivative, second derivative and variance for each sample in classes 1,2 and 3.

with and without SII in the feature set. The data sets for Adult 1-3 consists of 15 days of that subjects personal glucose measurements, and Global 15 days and Global 45 days consists of glucose measurements from Adult 4-10.

We can see from Table 4.1 that the inclusion of SII in the feature set has little or no impact on the classification when the data set consists of personal measurements. But for global data sets, consisting of glucose measurements from different subjects, there is a significantly improvement when SII in included.

Initial test of the different methods

To start with we take a look at how the different split methods and control limits performs in a fault free situation. The SPE-values for each class and $\text{result}_{\text{final}}$ for each split method and each control limit type can be seen in Figure 4.2, 4.3, 4.4 and 4.5. With en exemption of M2-limits as the SPE-plots would be identical to the ones with CS-limits, but with a higher limit. The glucose measurements consists of 24 hours from the main scenario for Adult 1. Meals are after 2,8, and 15 hours and of 35, 65 and 75 g. A window length of 10 (50 minutes) is used and the glucose measurements are given in mmol/L.

Table 4.1: Results of including SII in the feature set for a window with length 10.

Data set	Without SII	With SII	Difference
Adult 1	62.58 67.86 73.14	80.01 57.48 57.18	0.82
Adult 2	73.60 48.47 82.67	76.67 43.13 82.92	-2.02
Adult 3	60.90 54.52 46.43	60.90 54.52 46.43	0
Global 15 days	64.65 56.80 72.22	68.68 67.59 71.67	14.27
Global 45 days	70.18 58.33 66.57	66.47 66.35 70.55	8.29

Figure 4.2: SPE-plot for Split by SPE with M1-limit.

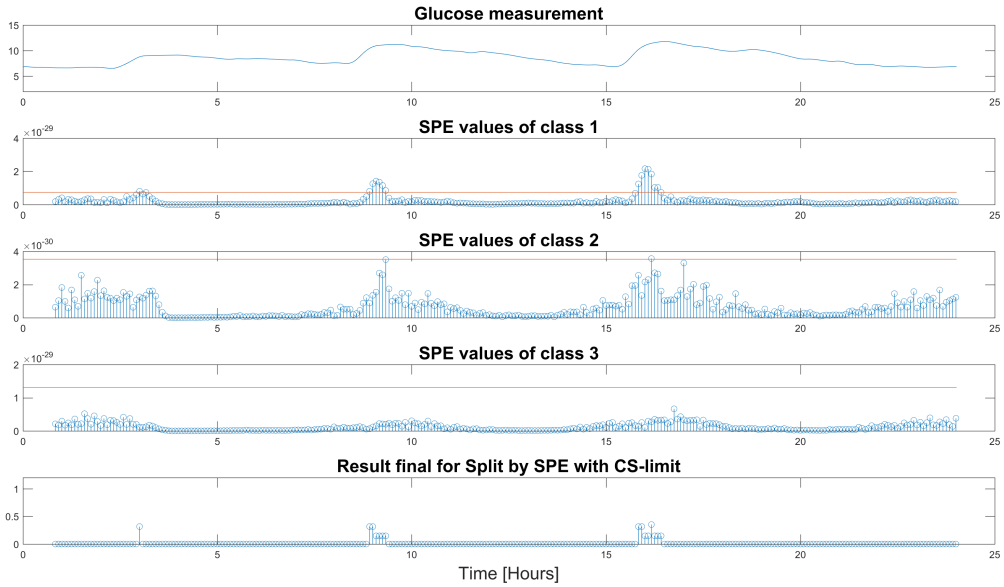

Figure 4.3: SPE-plot for Split by SPE with CS-limit.

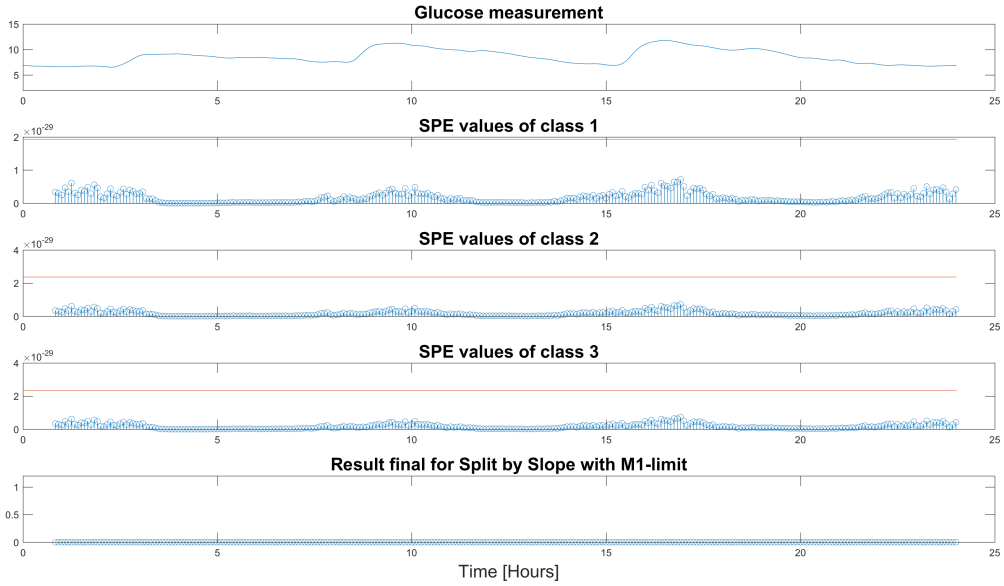

Figure 4.4: SPE-plot for Split by Slope with M1-limit.

As we can see the, different methods appear to be functioning as intended. At the same time we can see that the response differs for the different split methods. In Figure 4.2 and 4.3 we can se that split by SPE seems to be more sensitive than Split by Slope as there is a response in $\text{restult}_{\text{final}}$ even in the

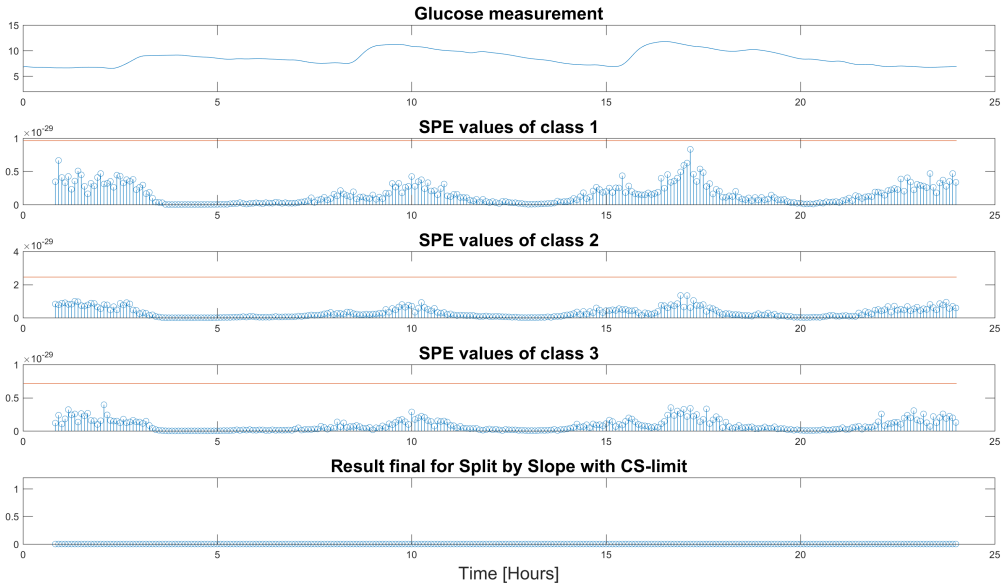

Figure 4.5: SPE-plot for Split by Slope with CS-limit.

fault free case. In Figure 4.2 we can see that Split by SPE combined with M1-limits is very sensitive and $\text{restult}_{\text{final}}$ is close to 1 several places. Therefore this method is dependent on a high fault limit to avoid false positives. In Figure 4.3 the response is smaller and might have been avoided if M2-limits are used instead of CS-limits. Split by Slope gives no indication of fault as seen in Figure 4.4 and 4.5. This can be a good sign as the glucose measurements are fault free, but at the same time it is possible that the split method gives low sensitivity. We can also see that the SPE-values and the final result do not start at the same time as the glucose measurements. This is because enough measurements are needed to fill the window before the calculations can be done. Another window length might be beneficial depending on the fault we are searching for, but a longer window gives a longer down period before the fault detection starts.

In Figures 4.6, 4.7, 4.8 and 4.9 the same scenario with the same meals are repeated, but this time a PISA is introduced after 12 hours. The PISA goes over 9 measurements (40 minutes) and has a depth of 5 mmol/L.

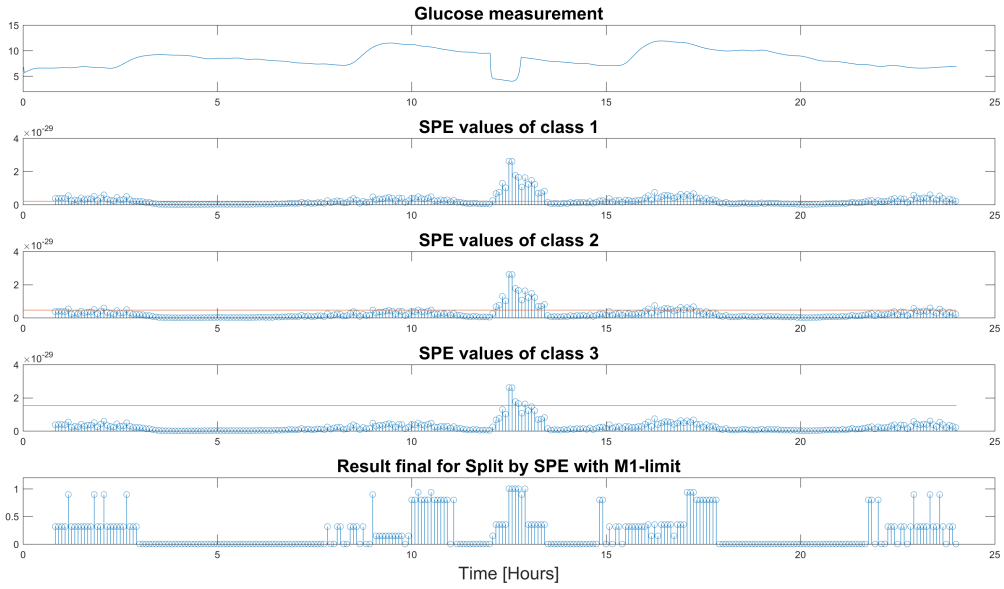

Figure 4.6: SPE-plot for Split by SPE with M1-limit and introduced PISA.

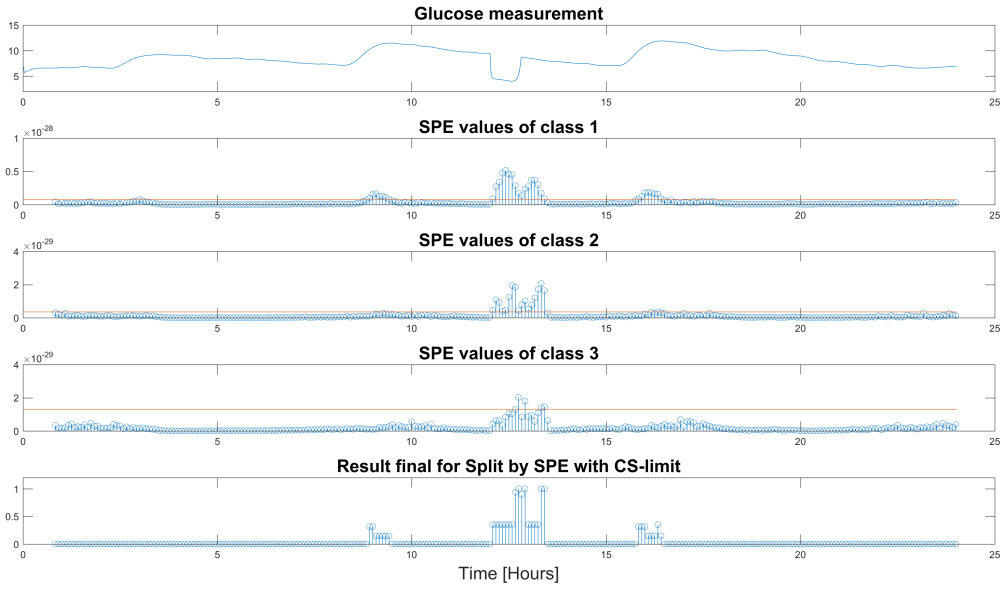

Figure 4.7: SPE-plot for Split by SPE with CS-limit and introduced PISA.

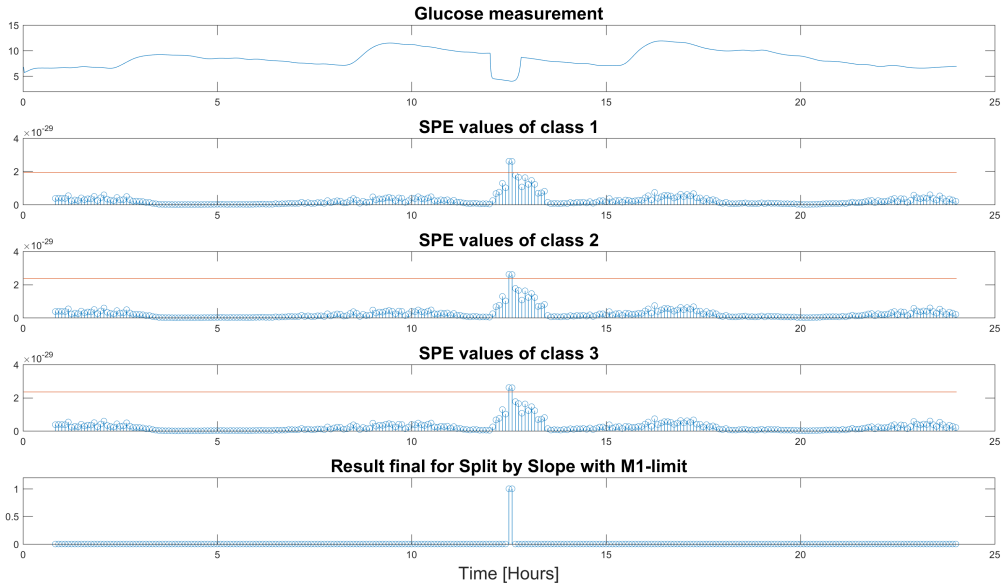

Figure 4.8: SPE-plot for Split by Slope with M1-limit and introduced PISA.

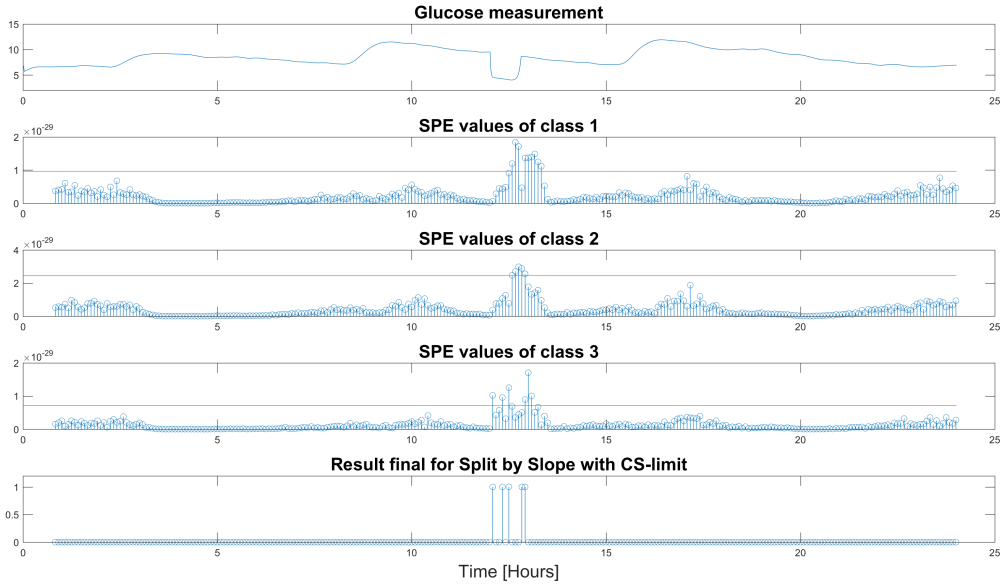

Figure 4.9: SPE-plot for Split by Slope with CS-limit and introduced PISA.

As we can see from Figure 4.6, 4.7, 4.8 and 4.9, all combinations of split methods and control limits gives a response to the fault. Interestingly the response is different between the methods which might indicate that the response is triggered by different glucose fluctuations. This may again mean

that different split methods and control limits might detect different types of fault with different precision. These plots are also only for a window of length 10. Different window lengths can again give us different responses. This leaves us with 6 permutations of split methods and control limits in addition to the variable window length, which gives us a good basis for fault detection.

4.1 Choice of main method and window lengths

Because of time constraints and the number of permutations of split methods, control limits and window lengths a simple approach was used to select the best combinations. Each split method was combined with each control limit and a number of chosen window lengths and tested on the main scenario. Each combination was simulated 10 times for loss of amplitude, PISA, infusion fault and spikes, and 10 times on fault free measurements. Jumping signal was omitted as implementation in the simulator made it difficult to define and count detections. This was first done for Adult 1 with personal data set consisting of 15 fault free simulations from the main scenario. It was then repeated for Adult 1, but this time with a global data set consisting of 45 fault free simulations from the main scenario from Adult 4-10. The faults was held constant and was the same for each of the 10 simulations. The only variables was meal times and amounts.

For loss of amplitude detection, detection time and the glucose concentration at the detection time was considered. Glucose concentration at the detection time was considered because it gives an indication of the rapidness of the detection. A detection is not usable if the signal has decreased too much as it might be dangerous for the user. For PISA detection, number of detections and detection time was considered. For infusion fault detection, detection time, and glucose concentration at the detection time was considered. Glucose concentration was considered on the same basis as with loss of amplitude. A detection of infusion fault is no good if the glucose concentration has risen to dangerous levels. For Spike the number of spikes detected was considered. The resulting tables are large and can therefore be found the attachment.

When selecting split methods, control limits and window lengths for further testing the selection was based on the combinations performance on each fault. The combinations with best performance for each fault are selected and used for fault detection. In addition the selected combinations must have a low number of false positives, as good fault detection is rendered useless if the user suffers from alarm fatigue. This resulted in three combinations was chosen for personal training, and three combinations was chosen for global training. The associated performance can be seen in Table 4.2 and 4.3. The column F/P contains the average number of false positives. For loss of amplitude there is percentage of detection, average detection time and average glucose concentration at detection. For PISA there is percentage of detection, average number of detections per PISA, and average detection time. For infusion fault

there is percentage of detection, average detection time and average glucose concentration at the detection. The last column shows the number of detected spikes.

Table 4.2: Performance of the selected methods, limits and window lengths for personal training.

Split method	Limit type	Window length	F/P	Loss of amplitude			PISA			Infusion fault			Spike
				Det.	Det. time	Det. G	Det.	Num. Det.	Det. time	Det.	Det. time	Det. G	Det.
SPE	M1	8	0.00	100 %	1199	4.25	67 %	4	34	100 %	209	17.01	0
SPE	M1	10	0.05	100 %	1220	3.59	80%	2	29	100 %	200	15.78	0
Slope	M2	10	0.45	100 %	1247	3.08	100 %	13	6	100 %	209	16.06	3

Table 4.3: Performance of the selected methods, limits and window lengths for global training.

Split method	Limit type	Window length	F/P	Loss of amplitude			PISA			Infusion fault			Spike
				Det.	Det. time	Det. G	Det.	Num. Det.	Det. time	Det.	Det. time	Det. G	Det.
SPE	M2	5	0.15	100%	1205	3.85	70%	2	20	100%	217	17.79	0
SPE	CS	3	0.10	100%	1253	2.49	0 %			100%	220	15.75	0
SPE	CS	5	0.05	100%	1255	2.52	100 %	1	14	100%	220	15.44	0

I total there is a difference in performance between the two split methods, and between personal and global training. For personal training Split by Slop combined with CS-limits performed well for all types of faults, but averages over one false positive per day. On the other side Split by SPE combined with M1-limit barely had a false positive, but with later detections. In general it seems that Split by Slope is better at detecting instantaneous faults as PISA, and that Split by SPE is better in regards to detecting loss of amplitude and infusion fault. Split by SPE also have fewer false positive than Split by Slope. For general training a lot of sensitivity was lost, both for Split by SPE and for Split by Slope. Split by Slope appears inferior to Split by SPE, as Split by Slope either have worse detection or more false positives.

The fact that none of the combinations is able to regularly detect spikes is a disappointment. Another interesting result is that none of the longer windows performs as good as the short ones, even on loss of amplitude that manifests over a long time.

4.2 Personal training

After the selections of split methods, limits and window lengths additional testing was done to find better fault limits. The final fault limits for the selection is given in Table 4.4.

Fault detection

For personal training the data set consists 15 fault free days from the main scenario with the current subject. The testing is done on three different subjects, Adult 1, Adult 2 and Adult 3. For each subject the main scenario is

Table 4.4: Fault limits for personal training.

Split method	Limit type	Window length	Fault limit
SPE	M1	8	90 %
SPE	M1	10	90 %
Slope	M2	10	100 %

simulated 10 times containing each fault. In addition 10 fault free days are simulated to test for false positives. The difference from the previous chapter is that the faults are now randomized. For detection of the jumping signal one window marked as a fault during a jump is classified as a detection of that jump. Detection time is not accounted for for loss of amplitude as the randomized faults renders the average detection time useless. Number of detections for PISA is not included as it obviously depend on the randomized PISA. The results for training and testing for Adult 1, Adult 2, and Adult 3 can bee seen in Table 4.5, 4.6 and 4.7.

Table 4.5: Fault detection results for Adult 1 with 15 days personal training.

Day	F/P	Loss of amplitude		PISA		Infusion fault			Spike	Jumping signal
		Det.	Det. G	Det.	Det. time	Det.	Det. time	Det. G	Det.	Det.
1	0	Yes	5.20	Yes	19	Yes	204	15.68	No	1/5
2	0	Yes	5.94	Yes	22	Yes	235	15.81	No	4/8
3	0	Yes	5.86	Yes	35	Yes	243	16.41	No	3/10
4	1	Yes	4.89	Yes	22	Yes	215	14.92	No	3/6
5	0	Yes	4.25	Yes	37	Yes	215	15.64	Yes	6/11
6	0	Yes	4.88	Yes	34	Yes	181	14.80	No	6/9
7	0	Yes	5.52	Yes	23	Yes	224	15.37	No	6/8
8	0	Yes	5.90	Yes	61	Yes	210	15.73	No	9/9
9	0	Yes	4.11	No		Yes	140	13.97	No	8/9
10	0	Yes	4.52	Yes	32	Yes	207	15.27	Yes	6/7
Total	0.1	100 %	5.1	90 %	32	100 %	207	15.36	20%	63 %

Table 4.6: Fault detection results for Adult 2 with 15 days personal training.

Day	F/P	Loss of amplitude		PISA		Infusion fault			Spike	Jumping signal
		Det.	Det. G	Det.	Det. time	Det.	Det. time	Det. G	Det.	Det.
1	3	Yes	3.69	Yes	10	Yes	212	14.99	No	3/3
2	0	Yes	4.66	Yes	8	Yes	167	14.24	No	6/6
3	0	Yes	3.62	Yes	13	Yes	160	14.64	No	9/9
4	0	Yes	4.80	Yes	8	Yes	180	14.73	No	5/6
5	0	Yes	4.91	Yes	12	Yes	210	14.92	No	9/11
6	1	Yes	3.66	Yes	8	Yes	180	15.75	No	8/9
7	1	Yes	4.05	Yes	6	Yes	197	14.45	No	4/5
8	0	Yes	3.98	Yes	9	Yes	214	14.79	No	8/9
9	0	Yes	3.25	Yes	7	Yes	212	14.85	No	7/9
10	0	Yes	3.26	Yes	7	Yes	216	14.81	No	4/5
Total	0.5	100 %	3.99	100%	9	100%	195	14.82	0 %	88 %

Table 4.7: Fault detection results for Adult 3 with 15 days personal training.

Day	F/P	Loss of amplitude		PISA		Infusion fault			Spike	Jumping signal
		Det.	Det. G	Det.	Det. time	Det.	Det. time	Det. G	Det.	Det.
1	0	Yes	4.54	Yes	12	Yes	153	14.17	No	$^{10}/_{10}$
2	0	Yes	3.84	Yes	11	Yes	185	14.29	No	$^{10}/_{10}$
3	1	Yes	3.93	Yes	127	Yes	216	13.78	No	$^{10}/_{10}$
4	0	Yes	6.20	Yes	11	Yes	183	14.26	No	$^{4}/_{4}$
5	0	Yes	3.30	Yes	15	Yes	173	13.49	No	$^{10}/_{10}$
6	0	Yes	3.92	Yes	17	Yes	225	14.68	No	$^{7}/_{7}$
7	0	Yes	3.33	Yes	12	Yes	236	15.19	No	$^{9}/_{9}$
8	0	Yes	3.47	Yes	18	Yes	209	15.32	No	$^{9}/_{10}$
9	0	Yes	2.86	Yes	10	Yes	222	15.77	No	$^{7}/_{7}$
10	0	Yes	3.72	Yes	7	Yes	181	14.05	No	$^{6}/_{6}$
Total	0.1	100 %	3.91	100 %	24	100%	198	14.50	0 %	99 %

As we can see from Table 4.5, 4.6 and 4.7 the three selected combinations for personal training is able to detect 4 out of the 5 faults and at the same time have an acceptable amount of false positives. The loss of amplitude is detected every time and is also detected long before the sensor signal is completely gone. The PISA is detected 29 out of 30 times and the short detection time is satisfactory as it gives little time for incorrect dosage of insulin. Infusion fault is detected every time, although ideally it could have been detected at a lower glucose concentration. Unfortunately very few of the spikes are detected, but that is not a surprise given it was rarely detected in testing done in 4.1. A positive result is the number of detections of the jumping signal as that fault was not part of the testing done when selecting combinations of split methods, control limits and window lengths.

4.3 Global training

As with the personal training additional testing was done to find better fault limits for the selected combinations. The resulting fault limits can be seen in table 4.8.

Table 4.8: Fault limits for global training.

Split method	Limit type	Window length	Fault limit
SPE	M2	5	70 %
SPE	CS	3	97 %
SPE	CS	5	97 %

4.3.1 45 day training set

For global training the training set consists of 45 fault free days from the main scenario with Adult 4-10. 6 days each and 1 day from three randomly selected subjects from Adult 4-10. This training set is then used with the selected split

methods, limits and window lengths, and tested on Adult 1, Adult 2 and Adult 3 in same fashion as for personal training. The results can be seen in table 4.9, 4.10 and 4.11.

Table 4.9: Fault detection results for Adult 1 with 45 days global training.

Day	**F/P**	**Loss of amplitude**		**PISA**		**Infusion fault**			**Spike**	**Jumping signal**
		Det	G	Det	t det	det	d time	G	det	det
1	0	Yes	3.63	Yes	12	Yes	200	14.39	No	4/7
2	1	Yes	3.64	Yes	31	Yes	141	13.39	Yes	4/6
3	0	Yes	3.48	No		Yes	156	14.05	No	4/10
4	0	Yes	3.32	Yes	11	Yes	180	13.84	No	7/11
5	1	Yes	1.11	No		Yes	186	14.49	No	3/6
6	0	Yes	3.15	No		Yes	205	14.05	No	7/10
7	4	Yes	3.83	No		Yes	224	14.65	No	5/9
8	0	Yes	1.66	No		Yes	214	15.79	No	3/5
9	0	Yes	1.74	Yes	105	Yes	121	14.11	No	6/7
10	0	Yes	3.42	Yes	114	Yes	211	16.57	No	5/6
Total	0.6	100%	2.90	40 %	55	100%	184	14.53	10%	62 %

Table 4.10: Fault detection results for Adult 2 with 45 days global training.

Day	**F/P**	**Loss of amplitude**		**PISA**		**Infusion fault**			**Spike**	**Jumping signal**
		Det	G	Det	t det	det	d time	G	det	det
1	0	Yes	4.04	Yes	21	Yes	207	14.50	No	6/7
2	0	Yes	3.39	Yes	10	Yes	184	14.06	No	5/8
3	0	Yes	2.88	No		Yes	160	13.62	No	4/8
4	0	Yes	2.82	Yes	13	Yes	151	13.58	No	5/7
5	0	Yes	2.40	Yes	9	Yes	215	14.74	No	4/7
6	0	Yes	3.45	Yes	18	Yes	169	13.19	No	7/9
7	0	Yes	2.60	No		Yes	190	14.96	No	7/9
8	0	Yes	3.01	Yes	41	Yes	187	15.48	No	5/7
9	0	Yes	2.41	Yes	12	Yes	179	14.55	No	6/8
10	0	Yes	3.34	Yes	13	Yes	191	13.99	No	7/7
Total	0	100 %	3.03	80%	17	100%	183	14.27	0 %	73 %

Table 4.11: Fault detection results for Adult 3 with 45 days global training.

Day	**F/P**	**Loss of amplitude**		**PISA**		**Infusion fault**			**Spike**	**Jumping signal**
		Det	G	Det	t det	det	d time	G	det	det
1	0	Yes	3.12	No		Yes	234	14.62	No	8/8
2	0	Yes	2.53	No		Yes	235	14.54	No	6/8
3	0	Yes	2.86	Yes	79	Yes	210	15.12	No	5/9
4	0	Yes	1.93	No		Yes	226	14.20	No	5/6
5	0	Yes	3.38	No		Yes	209	15.28	No	6/7
6	0	Yes	2.99	Yes	13	Yes	235	15.42	No	4/4
7	0	Yes	2.95	No		Yes	238	15.19	No	5/6
8	0	Yes	2.17	No		Yes	125	13.20	No	4/6
9	0	Yes	2.89	No		Yes	177	13.84	No	8/8
10	0	Yes	3.17	Yes	42	Yes	222	14.17	No	9/10
Total	0	100%	2.80	30 %	45	100%	211	14.56	0 %	83 %

As can be expected the results are not as good as with personal training. The detection of loss of amplitude is slowed down as the detection happens at lower glucose concentrations. Fewer of the PISA's are detected and with a higher detection time. When it comes to the detection of infusion fault the results are actually better than for the personal training, as the detection happens at a lower glucose concentration.

In total the main difference between the personal training and the global training us the slower detection of loss of amplitude and fewer detections of PISA. One possible cause might be that the large data set makes the selected combinations "blunter" and not able to detect loss of amplitude and PISA the same way as with a small data set. In addition it would be interesting to see if a change in training set size results in a change in fault detection performance. Therefore, a new try is made for global training with a data set containing only 15 simulations of the main scenario.

4.3.2 15 days training set

This time the global training set consists of 15 days from the main scenario with Adult 4-10, and is of the same size as the personal training data. 2 days each and 1 day from one randomly selected subject from Adult 4-10. It is also simulated independent of the 45 days used in the previous training set. The testing is performed in a similar fashion as the previous sections and the results can be seen in table 4.12, 4.13 and 4.14.

Table 4.12: Fault detection results for Adult 1 with 15 days global training.

Day	**F/P**	**Loss of amplitude**		**PISA**		**Infusion fault**			**Spike**	**Jumping signal**
		Det	G	Det	t det	det	d time	G	det	det
1	1	Yes	3.40	No		Yes	189	14.16	No	1/7
2	1	Yes	3.91	No		Yes	230	14.39	No	4/6
3	1	Yes	2.93	No		Yes	146	14.65	No	3/5
4	0	Yes	3.27	Yes	10	Yes	215	13.63	No	5/7
5	5	Yes	2.33	Yes	16	Yes	228	14.23	No	5/7
6	2	Yes	2.71	No		Yes	197	13.54	No	4/7
7	0	Yes	2.55	Yes	9	Yes	164	15.37	No	5/7
8	0	Yes	3.67	Yes	40	Yes	212	14.56	No	7/8
9	4	Yes	3.58	No		Yes	205	13.83	No	9/11
10	0	Yes	2.39	No		Yes	169	14.93	No	4/6
Total	1.4	100 %	3.08	40%	19	100%	196	14.33	0 %	66 %

Table 4.13: Fault detection results for Adult 2 with 15 days global training.

Day	F/P	Loss of amplitude		PISA		Infusion fault			Spike	Jumping signal
		Det	G	Det	t det	det	d time	G	det	det
1	0	Yes	2.76	No		Yes	176	13.96	No	1/5
2	0	Yes	2.82	No		Yes	150	12.99	Yes	3/7
3	0	Yes	2.91	Yes	39	Yes	169	15.76	No	4/7
4	0	Yes	2.27	No		Yes	176	13.39	No	6/6
5	0	Yes	3.30	No		Yes	176	13.10	No	6/11
6	0	Yes	2.91	Yes	18	Yes	144	14.60	No	4/5
7	0	Yes	3.04	No		Yes	177	13.17	No	8/9
8	0	Yes	2.63	Yes	47	Yes	164	13.33	No	8/9
9	1	Yes	2.13	Yes	12	Yes	198	15.00	No	4/5
10	0	Yes	3.00	No		Yes	188	14.25	No	9/9
Total	0.1	100 %	2.78	40 %	29	100%	172	13.96	10%	73 %

Table 4.14: Fault detection results for Adult 3 with 15 days global training.

Day	F/P	Loss of amplitude		PISA		Infusion fault			Spike	Jumping signal
		Det	G	Det	t det	det	d time	G	det	det
1	0	Yes	1.17	No		Yes	176	14.25	No	6/10
2	0	Yes	2.92	Yes	72	Yes	201	13.81	No	6/8
3	0	Yes	2.30	No		Yes	199	13.45	No	5/6
4	0	Yes	3.45	No		Yes	218	13.53	No	5/8
5	0	Yes	3.12	Yes	15	Yes	180	13.61	No	5/8
6	0	Yes	1.93	No		Yes	229	14.23	No	4/5
7	0	Yes	2.95	Yes	13	Yes	225	14.15	No	5/7
8	0	Yes	3.58	Yes	43	Yes	203	12.95	No	5/6
9	0	Yes	3.13	Yes	16	Yes	211	13.85	No	6/6
10	0	Yes	3.22	Yes	16	Yes	199	14.30	Yes	5/5
Total	0	100 %	2.78	60 %	29	100%	204	13.81	10 %	75 %

As can be seen in Table 4.12, 4.13 and 4.14, the fault detection results for global training data consisting of 15 days is close to identical to the results for the training data containing 45 days. The only significant difference is in the increased number of false positives for Adult 1. All loss of amplitude is detected at roughly the same glucose concentration. Only one less PISA was detected, all infusion faults was detected, and about the same number of jumps was detected. This shows that 15 days of glucose measurements is sufficient training data for the fault detection. It also indicates that the size of the training data not nececcarly correlated with the fault detection performance.

Detection of low glucose concentration

In addition to detection faults it would be interesting to see if the selected combinations are able to detect gradually decreasing glucose concentration, and in so be able to detect hypoglycaemia. And if so, at what glucose concentration is it detected. Therefore the implementation of the amplitude fault is exploited. A time period of 5 days (120 hours) is simulated with the fault. There is no meals and glucose concentration is supposed to be constant with exception of the noise. But the loss of amplitude will slowly decrease the

glucose concentration. This way we can see if the selected combinations will detect low glucose concentration that appear slowly. The results for personal training can be seen in Table 4.15, the results for global training with a large data set can be seen in Table 4.16, and the results for global training with a small data set can be seen in Table 4.17.

Table 4.15: Results for detection of low glucose concentration with 15 days personal training.

Day	**Adult 1**	**Adult 2**	**Adult 3**
1	5.63	4.89	5.89
2	6.00	5.18	5.23
3	5.73	5.08	5.02
4	5.82	4.89	5.75
5	5.81	5.26	5.49
6	5.95	5.09	5.74
7	5.89	4.92	5.69
8	5.97	4.92	5.40
9	5.73	4.98	5.56
10	5.79	5.14	5.84
Avg	5.83	5.04	5.56

Table 4.16: Results for detection of low glucose concentration with 45 days global training.

Day	**Adult 1**	**Adult 2**	**Adult 3**
1	4.54	4.81	4.14
2	4.51	4.66	4.34
3	4.45	4.01	4.90
4	4.74	4.39	4.38
5	4.22	4.55	4.14
6	3.85	4.44	4.69
7	4.57	4.82	4.56
8	4.20	4.06	4.62
9	4.53	3.98	4.15
10	4.03	4.44	4.12
Avg	4.36	4.41	4.40

In Tables 4.15, 4.16 and 4.17 we can see that the selected combinations is sensitive for gradually decreasing glucose concentration. For the personal training the thresholds vary, which indicates that the thresholds is depends on the training data. For the global training they are approximately the same for Adult 1-3. Interestingly the threshold for global training with the small data set is about the same as for the global training with a large data set.

Table 4.17: Results for detection of low glucose concentration with 15 days global training.

Day	Adult 1	Adult 2	Adult 3
1	3.97	4.61	4.28
2	4.27	4.32	4.32
3	4.17	4.28	4.34
4	4.02	4.20	4.18
5	4.35	4.15	3.91
6	4.21	4.27	4.51
7	4.40	4.38	4.29
8	3.75	3.96	4.32
9	4.19	4.02	4.01
10	3.85	4.17	4.28
Avg	4.12	4.24	4.24

Again we see that the size of the data set not necessarily have an effect on the performance of the selected combinations.

4.4 Fault recognition

In addition to detecting when a fault occur it needs to be decided what fault has occurred. The procedure proposed in section **??** is used and the window sets are used to train the classifier in 3.7. 200 windows for each of the selected combinations for fault is used. Obliviously it would be hard to get 200 detections of each fault for personal training, and a general fault recognition is needed also for personal training. Therefore Adult 1 with the corresponding personal training set of 15 days was used to make a training set for fault recognition for personal use. The fault recognition was then tested with Adult 2 and 3. For global training the data set consisting of 45 days was used and faults was detected for Adult 4-10. Therefore, the fault recognition is independent of Adult 1-3 on which it is tested on.

New simulations with unknown faults are run and when a fault is detected the classifier is used to decide what fault has occurred. Spikes are not included as they rarely are detected. Only the three first detections of a fault is considered as those are the most critical. It is assumed that most people will check the alarm in the first 15 minutes. Only the faults that was detected three or more times are taken into account in the following tables. The results for the personal training can be seen in Table 4.18 and 4.19, and the results for global training can be seen in Table 4.20, 4.21 and 4.22.

Table 4.18: Fault recognition results for Adult 2 with 15 days personal training.

Day	Loss of Amplitude	PISA	Infusion Fault	Jumping signal
1	1/3	3/3	0/3	3/3
2	3/3	3/3	3/3	2/3
3	0/3	3/3	3/3	2/3
4	0/3	3/3	1/3	3/3
5	0/3	3/3	3/3	1/3
6	2/3	2/3	3/3	0/3
7	3/3	3/3	1/3	2/3
8	2/3	3/3	3/3	0/3
9	0/3	3/3	3/3	0/3
10	2/3	2/3	0/3	0/3
Result	43 %	93%	67%	43%

Table 4.19: Fault recognition results for Adult 3 with 15 days personal training.

Day	Loss of Amplitude	PISA	Infusion Fault	Jumping signal
1	3/3	3/3	0/3	3/3
2	2/3	3/3	3/3	3/3
3	2/3	3/3	3/3	3/3
4	0/3	3/3	0/3	3/3
5	1/3	1/3	0/3	3/3
6	1/3	3/3	3/3	3/3
7	1/3	3/3	3/3	2/3
8	0/3	3/3	3/3	3/3
9	2/3	3/3	3/3	2/3
10	2/3	3/3	2/3	3/3
Result	47 %	93 %	67%	93%

Despite the fault recognition classifier being trained on windows detected based on another data set it preforms surprisingly well. For the personal training it classifies almost all PISA's correctly. There are some uncertainties like loss of amplitude being confused with PISA, which might indicate that the classifier is favouring PISA. Also infusion fault and jumping signal is occasionally confused. The fault recognition classifier for the global training data seems to do the opposite, and favour loss of amplitude over PISA. The weak classification of PISA is slightly disappointing, but most of the other classifications are on point, which is impressing with regards to the independent data set. As a whole the fault recognition performs reasonably well both for personal training and for global training, and we are able to decide what fault have occurred with some degree of accuracy.

Table 4.20: Fault recognition results for Adult 1 with 45 days global training.

Day	Loss of Amplitude	PISA	Infusion Fault	Jumping signal
1	3/3	1/3	3/3	1/3
2	3/3	1/3	3/3	3/3
3	3/3	1/3	3/3	3/3
4	3/3	1/3	3/3	3/3
5	3/3	0/3	3/3	3/3
6	3/3	1/3	3/3	1/3
7	2/3	2/3	1/3	1/3
8	3/3	1/3	0/3	3/3
9	3/3	1/3	3/3	0/3
10	3/3	1/3	3/3	3/3
Result	97 %	33 %	83%	70%

Table 4.21: Fault recognition results for Adult 2 with 45 days global training.

Day	Loss of Amplitude	PISA	Infusion Fault	Jumping signal
1	3/3	2/3	3/3	1/3
2	3/3	0/3	1/3	3/3
3	3/3	1/3	3/3	1/3
4	3/3	1/3	2/3	1/3
5	3/3	2/3	1/3	0/3
6	2/3	3/3	3/3	1/3
7	3/3	0/3	3/3	3/3
8	3/3	0/3	3/3	3/3
9	3/3	2/3	3/3	3/3
10	3/3	2/3	0/3	3/3
Result	97 %	43%	73 %	63%

4.5 Conclusion

The aim of this thesis was to develop a fault detection method able to detect several types of faults. The work of [10] was used as a starting point, and a similar method was developed and implemented in the APT-simulator. Two new methods for splitting the training data was proposed. Both functioned as intended and was usable for fault detection. For personal training, each split method had strengths and weaknesses. Split by SPE gave the best results for detection of loss of amplitude and infusion fault, while Split by Slope gave the best results for detection of PISA. In the case of global training, Split by SPE outperformed Split by Slope with regards to all faults. Inclusion of IIR was another new function. The effect was negligible for personal training, but for global training, a clear positive effect was observed for the classification of windows.

Table 4.22: Fault recognition results for Adult 3 with 45 days global training.

Day	Loss of Amplitude	PISA	Infusion Fault	Jumping signal
1	3/3	1/3	3/3	0/3
2	3/3	1/3	3/3	1/3
3	3/3	2/3	3/3	1/3
4	3/3	1/3	3/3	3/3
5	3/3	1/3	3/3	3/3
6	3/3	1/3	3/3	0/3
7	3/3	2/3	3/3	0/3
8	3/3	1/3	3/3	2/3
9	3/3	1/3	3/3	2/3
10	3/3	1/3	1/3	2/3
Result	100 %	40%	93%	47%

The implementation of several windows at the same time strengthens the system as no window performs well for all faults. The possibility of using several windows then enabled us to use windows which performed well for different faults. The fault detection was tested for both personal training and global training. As can be expected, the personal training performed better than the global, but the chosen combinations of split methods, control limits, and window lengths, was able to detect faults in both cases. In addition to detecting faults, a fault recognition procedure was constructed and applied. The constructed system was able to classify the detected faults in a manner which produced decent results.

In conclusion, we have proposed and implemented a working fault detection and recognition system. The method has an acceptable number of false positives, is able to detect 5 different faults, and is sensitive to low glucose concentration. Additionally, the method is capable of differentiating between the 4 faults. The described fault detection method has potential, and can be improved with further work. It may very well serve as a building block for a method that is capable of to detecting a range of different types of faults, and is a step towards a fault detection method that can be used in an real artificial pancreas.

Chapter 5

Suggestions for future work

For the current implementation there is some room for tuning of different parameters, such as the fault limits, that might boost the performance. The chosen combinations of split methods, control limits, and window lengths may not be optimal, and a different combination might give better results.

The largest potential for improvement lays in the splitting of the training data into classes, which is the corner stone for this fault detection. The current methods works, but is simplistic. A more sophisticated method for splitting the data that results in better PCA-models for each class will have several positive effects. First of all the improved PCA-models will have a lower average SPE-value within each class, which will result in a more sensitive fault detection. A more sensitive and sharp detection will also make it easier to differentiate between different types of faults. A sensible splitting method might also make classification of new unknown windows easier as with Split by Slope. Another approach might be to split the training data into more than three classes. It might improve the PCA-models and further make the fault detection more sensitive. But at the same time it will make classification of new windows more difficult. It is a trade off, but one worth examining. Especially if the split method of the training data is easy and classification is easy. The split method and number of classes also effects the performance of the windows used in the fault detection. A new split method and a different number of classes might enable us to use other window lengths and use specific windows for specific faults. And as a result an entirely different combination of window lengths might be beneficial.

The classifier used is linear and another type of classifier might improve the results, both for the fault detection and for the fault recognition. There are many different directions one can take, but maybe the most interesting is maybe artificial intelligence. A artificial neural network might improve the classification of new windows, and what could fit better to an artificial pancreas than artificial intelligence? On the other hand the progression from a linear to a non-linear classifier is not without challenges and one need to avoid for example overfitting.

One drawback with this fault detection method is that it is only tested on faults implemented in the APT-simulator. The fault detection method

should be tested on other fault implementations to see if one could produce comparable results. The method is also solely tested on simulated data, so testing on real life data should also be a natural next step.

References

[1] Nihat Baysal, Fraser Cameron, Bruce A Buckingham, Darrell M Wilson, and B Wayne Bequette. Detecting sensor and insulin infusion set anomalies in an artificial pancreas. In *American Control Conference (ACC), 2013*, pages 2929–2933. IEEE, 2013.

[2] Nihat Baysal, Fraser Cameron, Bruce A Buckingham, Darrell M Wilson, H Peter Chase, David M Maahs, B Wayne Bequette, and In Home Closed-Loop Study Group (IHCL). A novel method to detect pressure-induced sensor attenuations (pisa) in an artificial pancreas. *Journal of diabetes science and technology*, 8(6):1091–1096, 2014.

[3] Simone Del Favero, Marco Monaro, Andrea Facchinetti, Alessia Tagliavini, Giovanni Sparacino, and Claudio Cobelli. Real-time detection of glucose sensor and insulin pump faults in an artificial pancreas. *IFAC Proceedings Volumes*, 47(3):1941–1946, 2014.

[4] Andrea Facchinetti, Simone Favero, Giovanni Sparacino, and Claudio Cobelli. An online failure detection method of the glucose sensor-insulin pump system: improved overnight safety of type-1 diabetic subjects. *IEEE Transactions on Biomedical Engineering*, 60(2):406–416, 2013.

[5] Daniel P Howsmon, Faye Cameron, Nihat Baysal, Trang T Ly, Gregory P Forlenza, David M Maahs, Bruce A Buckingham, Juergen Hahn, and B Wayne Bequette. Continuous glucose monitoring enables the detection of losses in infusion set actuation (lisas). *Sensors*, 17(1):161, 2017.

[6] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[7] Zeinab Mahmoudi, Dimitri Boiroux, Morten Hagdrup, Kirsten Nørgaard, Niels Kjølstad Poulsen, Henrik Madsen, and John Bagterp Jørgensen. Application of the continuous-discrete extended kalman filter for fault detection in continuous glucose monitors for type 1 diabetes. In *Control Conference (ECC), 2016 European*, pages 714–719. IEEE, 2016.

[8] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34, 2014.

[9] Paul Nomikos and John F MacGregor. Multivariate spc charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995.

[10] Guangjian Song and Chunhui Zhao. An effective fault detection method with fda classifier and global model for continuous glucose monitor (cgm). In *Control Conference (CCC), 2017 36th Chinese*, pages 7448–7453. IEEE, 2017.

[11] Guangjian Song, Chunhui Zhao, and Youxian Sun. A classification-based fault detection method for continuous glucose monitoring (cgm). In *Intelligent Control and Automation (WCICA), 2016 12th World Congress on*, pages 956–961. IEEE, 2016.

[12] Øyvind Stavdahl, Anders L Fougner, Konstanze Kölle, Sverre Chr Christiansen, Reinold Ellingsen, and Sven M Carlsen. The artificial pancreas: a dynamic challenge. *IFAC-PapersOnLine*, 49(7):765–772, 2016.

[13] Kamuran Turksoy, Elif S Bayrak, Lauretta Quinn, Elizabeth Littlejohn, Derrick Rollins, and Ali Cinar. Hypoglycemia early alarm systems based on multivariable models. *Industrial & engineering chemistry research*, 52(35):12329–12336, 2013.

[14] Kamuran Turksoy, Iman Hajizadeh, Elizabeth Littlejohn, and Ali Cinar. Multivariate statistical monitoring of sensor faults of a multivariable artificial pancreas. *IFAC-PapersOnLine*, 50(1):10998–11004, 2017.

[15] Kamuran Turksoy, Laurie Quinn, Elizabeth Littlejohn, and Ali Cinar. Monitoring and fault detection of continuous glucose sensor measurements. In *American Control Conference (ACC), 2015*, pages 5091–5096. IEEE, 2015.

[16] Robert A van den Berg, Huub CJ Hoefsloot, Johan A Westerhuis, Age K Smilde, and Mariët J van der Werf. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC genomics*, 7(1):142, 2006.

[17] Chunhui Zhao and Yongji Fu. Automatic and online fault detection of sensor problems using continuous glucose monitoring data for type 1 diabetes. In *Control Conference (CCC), 2014 33rd Chinese*, pages 3181–3186. IEEE, 2014.

[18] Chunhui Zhao and Yongji Fu. Statistical analysis based online sensor failure detection for continuous glucose monitoring in type i diabetes. *Chemometrics and Intelligent Laboratory Systems*, 144:128–137, 2015.