Timo Karlsson

Indirect Detection of Ice on Wind Turbine Rotor

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 10.1.2013

Thesis supervisor:

Prof. Kai Zenger

Thesis instructor:

M.Sc. Tomas Wallenius

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Timo Karlsson

Title: Indirect Detection of Ice on Wind Turbine Rotor

Date: 10.1.2013 Language: English Number of pages:6+81

Department of Automation and Systems Technology

Professorship: Control Engineering Code: AS-74

Supervisor: Prof. Kai Zenger

Instructor: M.Sc. Tomas Wallenius

Wind turbine icing causes issues for wind turbines in cold climates. Ice causes production losses and increases loads on turbine structures and components. Icing conditions can be identified with an ice sensor, but ice accretion on the turbine is harder to detect directly with a sensor. On the other hand, the effects icing has on turbine behaviour are known.

This thesis shows an approach for ice detection indirectly, by searching the turbine process data for signs of icing. This is done in real time by looking for abnormal values in standard process measurements using statistical methods.

Detection is done by using three different control charts, a principal component analysis -based method and a method based on k nearest neighbour search.

The effectiveness of these methods is examined in a simulation study. The sensitivity, accuracy and detection speeds of all five methods are compared in different ice and wind cases. Finally the methods are tested on authentic process data from a real wind turbine.

The methods introduced are able to react appropriately to changes in the data in the simulated test case, but the accuracy is dependant on wind speed. The methods do find probable icing incidents from real process data, but overall detection accuracy in real world use still leaves a lot of room for improvement.

Keywords: Wind turbine, Icing, Control charts, fault detection, Statistical process control

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Timo Karlsson

Työn nimi: Tuulivoimalan roottoriin kertyneen jään epäsuora havaitseminen

Päivämäärä: 10.1.2013 Kieli: Englanti Sivumäärä:6+81

Automaatio- ja systeemitekniikan laitos

Professuuri: Systeemitekniikka Koodi: AS-74

Valvoja: Prof. Kai Zenger

Ohjaaja: DI Tomas Wallenius

Tuuliturbiinin lapoihin kertyvä jää tuottaa vaikeuksia tuulivoimaloille kylmässä ilmastossa. Jää aiheuttaa tuotantotappioita ja kasvattaa turbiinin rakenteisiin ja komponentteihin kohdistuvia kuormia. Jäätävät olosuhteet voidaan tunnistaa, mutta jään olemassaoloa turbiinin lavoissa on vaikeampi mitata. Jäätämisen vaikutukset turbiinin toimintaan kuitenkin tunnetaan.

Tavoitteena on kehittää ja testata menetelmä, jolla voidaan havaita turbiiniin kertyvä jää tarkkailemalla turbiinissa normaalisti tehtäviä mittauksia. Turbiinin prosessidatasta etsitään jäätämiseen viittaavia, tavallisesta toiminnasta poikkeavia arvoja tilastollisten menetelmien avulla.

Sovelletut menetelmät ovat normaalisti laadunvalvonnassa käytettyjä tilastolliseen prosessinohjaukseen suunniteltuja menetelmiä, joita käytetään tässä työssä jään aiheuttamien ilmiöiden tunnistamiseen.

Menetelmien toimivuutta testataan ensin simulointimallin tulosten avulla erilaisia tuuli- ja jääolosuhteissa. Tämän jälkeen menetelmien toimivuutta kokeillaan myös oikealla tuulivoimalan prosessidatalla.

Työssä käytetyt menetelmät onnistuvat löytämään jäätymistapaukset simulointidatasta melko hyvin, tosin tunnistustarkkuus on riippuvainen tuulen nopeudesta. Mittaustulosten analysointi kuitenkin osioittaa, että tarkkuudessa on vielä parannettavaa.

Avainsanat: Tuulivoima, Jääntunnistus, Tilastollinen prosessinohjaus

Contents

Symbols and abbreviations

Symbols

λ	Tip-speed ratio of the turbine
r	Turbine blade length [m]
Ω	Rotational speed of the turbine rotor [rad/s]
P	Power [W]
ρ	density of air [kg/m^3]
v	wind speed [m/s]
C_p	Capacity factor (efficiency) of the turbine
b	bin width
i	bin index
B_i	Bin at index y (a set of measurements)
n_i	number of measurements in bin B_i
$x_j(t)$	process variable (measurement at time t)
$m_j(i)$	mean of measurements x_j in bin B_y
$s_j(i)$	standard deviation of measurements x_j in bin B_y
$\hat{x}_j(t)$	interpolated estimate of $x_j(t)$
$z_j(t)$	Normalised process variable $x_j(t)$
p	pressure [Pa]
T	temperature [$\circ C$]
R	ideal gas constant of dry air [J/molK]
v_n	normalised (temperature corrected) wind speed
h_r	cut-off limit for construction of reference dataset for kNN search
h_a	alarm limit for kNN-search
k	number of neighbouring values used in kNN algorithms
d	distance measure
r	smoothing parameter
d_e	Euclidean distance
d_{cb}	citiyblock distance
d_c	Chebychev distance
$\mathbf{X}$	measurement matrix
$\mathbf{x_j}$	vector containing measurements of variable x_j
$\mathbf{Z}$	rescaled $\mathbf{X}$
$\mathbf{z_j}$	rescaled $\mathbf{x_j}$
$\overline{x_j}$	mean of $\mathbf{x_j}$
σ_j	standard deviation of $\mathbf{x_j}$

$\mathbf{R}$	covariance matrix of $\mathbf{X}$
$\mathbf{p_n}$	principal component, eigenvector of $\mathbf{R}$
λ_n	eigenvalue of $\mathbf{R}$
$\mathbf{P}$	projection matrix, consist of principal components $\mathbf{p_n}$
$\mathbf{T}$	Principal component scores
$\tau_\mathbf{j}$	principal component score corresponding to measurement x_j
$\hat{\mathbf{X}}$	backprojection of $\mathbf{X}$
$\mathbf{E}$	error matrix in principal component projection
h_A	upper limit for ACR
T^2	Hotellings statistic
Q	Q-statistic corresponding to principal component projection
$\mathbf{r}$	residual vector
Q_α	upper limit for the Q_P statistic
$S(T)$	cumulative sum at time T
$S_H(t)$	cumulative sum
$S_L(t)$	cumulative sum
$\mathbf{R_E}$	scaling matrix for EWMA chart calculation
Q_E	EWMA control chart
h_E	upper limit for EWMA control chart
$\mathbf{\Sigma}$	process covariance matrix
$\mathbf{\Sigma_{z(t)}}$	estimate of $\mathbf{\Sigma}$ at time t
$\mathbf{x}(t)$	vector of measurements of all process variables at time t
$\mathbf{z}(t)$	normalised $\mathbf{x}(t)$
μ	vector containing the means of process variables x_j
D^2	Mahalanobis distance between $\mathbf{x}(t)$ and μ
$S_M(t)$	multivariate cumulative sum at time t

Abbreviations

FAST	Fatigue, Aerodynamics, Structures, and Turbulence; Simulation model used to generate the data
NREL	National Renewable Energy Laboratory; National laboratory of the United States Department of Energy
MSPCA	Multivariate statistical process control
CUSUM	cumulative sum of a single variable
MCUSUM	multivariate CUSUM
EWMA	exponentially weighted moving average control chart.
MEWMA	Multivariate EWMA chart
kNN	k nearest neighbours
PCA	principal component analysis
ACR	Accumulative contribution rate

1 Introduction

Wind turbine icing is a significant problem for wind power applications in cold climate conditions. Icing can cause fatigue in turbine components, shortening the lifetime of the turbine and can cause immediate production losses. Detecting ice directly is difficult to do in a reliable way. The aim of this work is to compare different methods that make it possible to detect icing indirectly from normal wind turbine process data.

For the purposes of this work icing is seen as an external fault in the turbine. Ice detection then becomes a question of detecting abnormal changes in turbine behaviour. All the methods introduced in this work operate solely based on historical process data without knowledge about the underlying system. The used methods are generic methods normally used for fault detection.

All the methods used here follow a similar two-step approach: First a baseline relationship between wind and different process variables is determined using historical data. Then real-time measurements are compared to the historical baseline. Possible icing events are identified by monitoring the differences between the predetermined baseline and the measurements. The comparison is done by calculating a distance measure i.e. a single number that indicates how well the most recent measurement corresponds to the baseline. This distance measure can then be used as an ice alarm signal.

The used methods are all statistical methods that do not make any assumptions about the exact nature of the observed system or the nature of the faults. The resulting system can then be used for different types of turbines without any real changes.

Ice detection requires that the short-term effects of icing on the wind turbine behaviour and the measured variables need to be identified first. These effects are deduced from analysing results from simulation studies using several wind distributions and multiple ice cases. Later these same simulations are used to test the effectiveness of the different ice detection methods. Finally the methods are tested on real-world wind turbine data. The tests with the turbine data also offer an opportunity to compare these indirect methods with a standard ice sensor.

2 Wind turbine icing

2.1 Wind Turbine

The basic structure of a modern day wind turbine is illustrated in Fig. 2.1. Turbine nacelle is rotated to face the wind. Turbine output power is controlled to achieve optimal power production for the current wind speed by altering the blade pitch angle i.e. the angle of the blade relative to the wind. The optimal output power is reached when the relative speed of the blade tip as compared to the wind speed is kept at a certain optimum. Optimal tip-speed ratio is a design feature of the turbine blades and is usually a constant. Tip-speed ratio is defined by (1) where (1) r is the blade length, v is wind speed and Ω is the rotational speed of the rotor in rad/s. [1]

$$\lambda = \frac{r\Omega}{v} \tag{1}$$

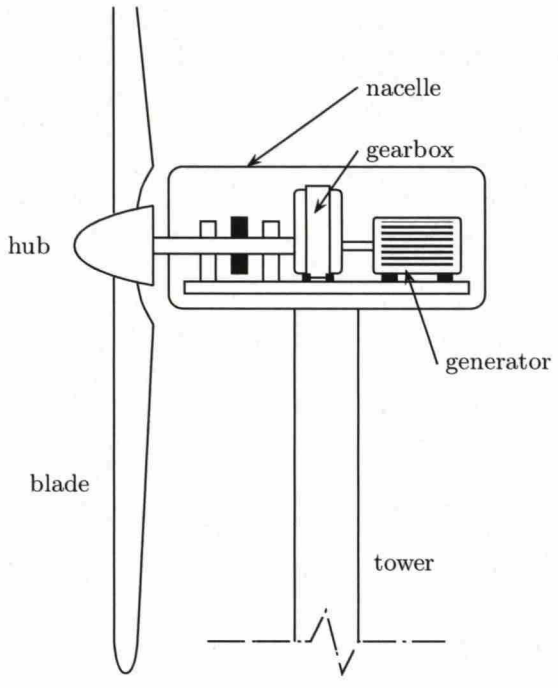

Figure 1: Basic structure of operational parts of a wind turbine [2]

All wind turbines have an optimal power curve i.e. a certain, predefined, optimal relationship between the wind speed and generator power. Power curve is usually defined for one set of conditions. Normally the generated power fluctuates around the power curve to a certain degree. Turbulence, temperature and air pressure all affect the produced output power at a set wind speed to a varying degree. The maximum amount of power that can be extracted from wind is defined by (2). [1]

$$P_w = \frac{1}{2}\rho\pi r^2 v^3 C_p(\lambda) \tag{2}$$

In (2) r is the blade length, v is the wind speed. ρ in (2) is air density. Air density on the other hand is directly proportional to air temperature and air pressure. $C_p(\lambda)$ is the so-called capacity factor, or turbine efficiency, that depends on the turbine tip-speed ratio λ. $C_p(\lambda)$ is turbine specific, the shape of the $C_p(\lambda)$-curve depends on blade aerodynamics. An example is illustrated in Fig.2.1. Figure 2.1 is for a two-bladed turbine, but the basic shape is very similar for a more modern three bladed design as well.

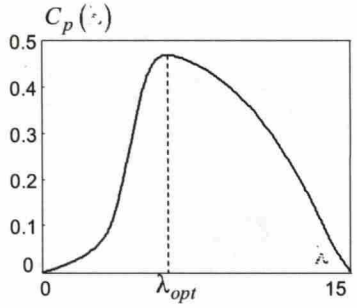

Figure 2: The basic shape of a $C_p(\lambda)$ -curve for a two-bladed turbine. [1]

Wind turbine power curve is represented by three wind speeds: the cut-in speed, the rated speed and the cut-off speed. The cut-in speed is the point where the turbine is started. It represents the windspeed at which the turbine is able to overcome its own power losses. The cut-out speed is the maximum wind speed at which the turbine is allowed to operate. If wind speed raises above the cut-out point, the turbine blades are turned into an angle that does not produce any rotational force and the rotor is stopped. In some cases the nacelle is also rotated away from wind after the turbine has been stopped to prevent accidental restarts. The third characteristic point on a power curve is the wind speed at which the turbine reaches its nominal (rated) power. Generator power increases along with windspeed up until this point. After the rated power point has been reached the turbine will operate at its nominal power as long as the wind speed stays above the rated power point and below the cut-out speed. [1]

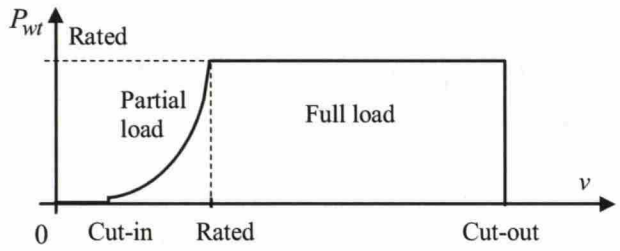

Figure 3: A power curve and operating regions of a wind turbine [1]

The power is kept constant at higher wind speeds by adjusting the angle of the turbine blades in order to weaken the aerodynamic properties of the turbine. As a

result the turbine will operate at a lower tip-speed ratio. Lower relative speed will result in drop in produced power as well.

2.2 Icing

When installing wind turbines in to locations where temperatures can drop below 0 °C, there is always a risk of icing. Icing can cause production losses and unwanted stops and in extreme cases even damage the turbines. Because of this in areas where icing is possible it needs to be taken into account and some kind of actions are needed. Icing risks can not be completely ignored when planning future wind power sites, because there is a very large wind power potential in areas affected by icing.

There are two types of icing: in-cloud icing and precipitation icing. In-cloud icing occurs when small droplets of liquid water that form clouds freeze when hitting a surface. These small water droplets can remain in liquid form in air in freezing temperatures due to the small size of individual droplets. They will only freeze when hitting a surface. Precipitation icing mostly refers to freezing rain or wet snow. [3]

Ice accumulates on all parts of the turbine, but the most problematic form of icing is the ice accumulated on the turbine blades. This has direct effects on the dynamics of the turbine and can cause production losses and increase fatigue loads in the turbine. Ice accretion on the blades increases the blade mass which in turn increases the loads in turbine structures. If ice is not distributed evenly, the mass imbalance can cause unwanted vibrations. Both of these can cause fatigue in turbine components which might shorten the lifetime of the turbine and can cause unwanted breakages. [4]

Even a small number of ice will have a noticeable effect on the output power of wind turbine. The detrimental effect will become more severe as ice mass increases but an ice sheet only a few mm thick can have noticeable effects on the amount of power a turbine is able to produce. [5]

Even if the ice accretion is not severe enough to cause problems through sheer mass alone the ice accumulated on the blades can have a detrimental effect on turbine dynamics through changes in blade aerodynamics. Ice accretion on a turbine blade increases the drag (air resistance) of the blade and decreases the lift produced by the blade. As a result an iced turbine blade will produce less power at a constant wind speed than a clean blade would. As a result of this the relationship between wind speed and the rotational speed of the turbine will change. The relative speed of the blade tip is a design parameter for the turbine, the ratio of blade tip speed to wind speed has an optimal value where the turbine produces maximum torque.[6],[7]

The change in lift and drag coefficients changes the torque produced by the turbine at a certain wind speed. This will in return cause the turbine to rotate at a slower speed when compared to wind speeds. On a variable pitch turbine this means that the pitch angle related to wind speed does not behave as it is supposed to. Pitch angle controller starts to alter the angle of attack at a certain rotational speed to prevent the turbine from spinning too fast. On an iced turbine this might

happen at a higher wind speed than it normally should. [2]

All these changes can be detected by monitoring the differences between standard behaviour and current values. The standard values can be deducted from process data under warm conditions, or set beforehand using predefined power, speed and blade angle curves. A more interesting approach is to collect process data in regular conditions first. Using old process data makes it possible to react to variation in the actual values of the process variables. Wind is never constant and the available wind measurement is very likely to be corrupted to some degree because wind is often measured with an anemometer located on top of the nacelle. This means that there will be large fluctuations in available measurements, and the used method needs to be able to react to this.

2.3 Ice sensors

A normal ice sensor solution is to install an ice sensor on top of the turbine nacelle. This can be problematic due to sheer size of a modern turbine. It is very well possible that a turbine blade might be covered by a cloud, even when the sensor isn't. Often an ice sensor is more of a point-like detector, it only reliably detects icing at the sensor location. Because of this an ice sensor is more an indicator of icing conditions than an indicator of presence of ice. Conditions on top of the nacelle are not equivalent to conditions at the blade tip. Most importantly the apparent wind speed at the blade tip is significantly (approximately 7 times) higher when the turbine is operating. Faster relative speed and smaller impact surface (only the narrow leading edge of the blade) means that the cooling effect of the surrounding air is amplified significantly at the blade tip.

The best results could be achieved by mounting the ice detector on the blade itself. This is technically very demanding, the sensors need electricity and signal cabling, and are extremely hard to replace in the case of malfunctioning sensor. The blade isn't completely rigid either and the surface is made to be as smooth and slippery as possible to make it as aerodynamically efficient as possible. Even if a sensor is installed on a blade, the sensor would still need to be able to detect icing over a larger area, simply because of the sheer size of the turbine blade. Ice can accumulate on different parts of the blade and a sensor can usually only cover a part of the blade. [3]

The blade tip is the most important target, because the very tip of the blade produces a significant portion of the power in the turbine in the first place. According to [7], ice on the outmost 5 % of the blade has an equally large detrimental effect on turbine performance as ice on the inner 95 %. In addition to this only ice on the last 25 % has a real effect on turbine performance.

Many ice sensors detect changes caused by ice accumulated on the sensor itself or some kind of detector part such as a probe, thin film or a wire. The most common operating principle for an ice sensor is to monitor a change in some property caused by icing. The property can be an electrical property of the probe such as capacitance, inductance or impedance. Or it can be the amplitude of an ultrasonic wave or simply ice mass accumulated on the detector. [3]

Because of these issues with ice sensors, indirect detection is an attractive solution. It does not need extra equipment, which makes it significantly easier to add detection capability to an already operating turbine. The approach used in this work also tries to detect actual consequences of icing rather than issuing warnings because the conditions are right. The methods used in this work are reactive, which means that they will issue icing alarms only after there is enough ice to have an effect on performance on the blades.

3 Simulation study

A controlled dataset is needed to reliably test all the different methods. This minimizes possible problems caused by measurement issues and makes algorithm testing somewhat easier. FAST (Fatigue, Aerodynamics, Structures, and Turbulence) is a complete simulation model for a wind turbine system. It uses a multibody dynamics approach and models the turbine as a combination of rigidly connected flexible bodies. The model of a wind turbine used by the simulation model consists of a few main components as illustrated in Fig. 2.1. The model makes it possible to study e.g. turbines aerodynamic behaviour in changing conditions and the forces affecting the turbine structure. FAST has been certified by Germanischer Lloyd WindEnergie and found suitable for "the calculation of onshore wind turbine loads for design and certification". [8] [9]

The turbine used in the simulations is a 5 MW reference turbine specified by NREL in [10]. It is a reference turbine model meant to be used as a baseline for wind turbine simulations. The NREL reference turbine is a model built based on publicly available information about real production turbines. It is a composite system mostly based on existing RePower 5MW turbine.

The biggest limitation in using FAST is that it does not support changing the turbine dynamics mid-run. This means that simulations do not cover the somewhat interesting icing event, but can simulate the behaviour of an already iced turbine and compare this to a turbine behaviour in normal conditions. This is enough to see the changes caused by icing but it makes evaluating the sensitivity of any method used to detect icing difficult. It is possible to see the differences between different icing scenarios with different ice masses and it is possible to make estimates on the detection speed of used methods. But in crease in ice mass in these simulations will be a step-like process.

The step transition is not realistic, but on the other hand it makes assessing the detection speed and accuracy easier. On these datasets the exact moment when the icing event starts or stops is known exactly. For testing purposes, several datasets with different ice masses were created. Then, when running the simulation with identical wind time series, the differences in turbine behaviour are all caused by blade icing, because all other variables are identical and all the differences in process variables between simulation runs are caused by differing ice masses.

The icing cases used in simulations are collected in Table 1. The approximate shapes of ice accumulated an the turbine blade in different cases are illustrated in

Fig. 3. Figure 3 represents an approximate shape of ice on the tip of the turbine blade.

Table 1: The ice masses used in simulations [11]

case	name	ice mass	description
1	start of icing	0.2 kg / m	Ice appears on blade leading edge. Maximum thickness approximately 0.2 cm on approximately 85 % of the blade length.
2	light icing	2 kg / m	Ice appears on blade leading edge. Maximum thickness 2 cm on approximately 85 % of the blade length.
3	moderate icing	8 kg / m	Ice appears on blade leading edge. Maximum thickness 10 cm on approximately 85 % of the blade length.

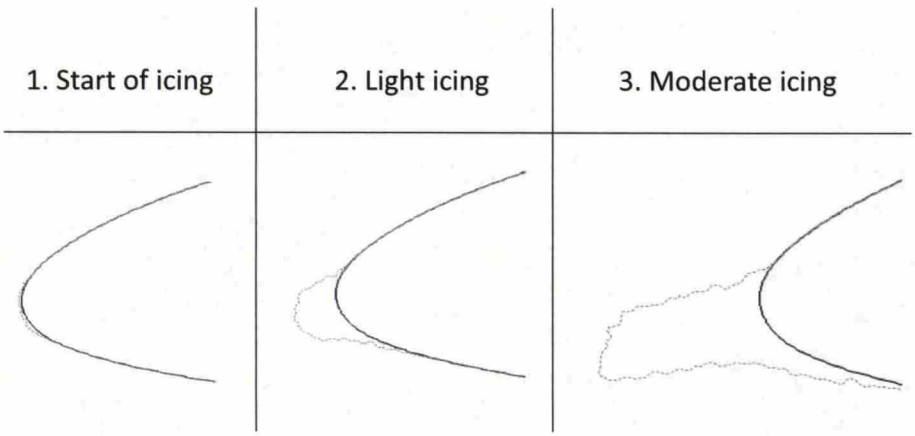

Figure 4: Ice shapes in different ice cases [11]

The simulations are made with three different ice masses and with symmetric and asymmetric icing. Asymmetric icing here means that one blade of a three blade turbine is clean and the remaining two blades are iced identically. Asymmetric icing is a more severe scenario because it causes unbalance in aerodynamic forces and blade masses. This will increase loads on turbine structures. [12] [13]

To test the methods several reference datasets were created. Time series plots of these test cases are illustrated in Figs. 5 and 6. Simulations have been run for each time series for each ice case in table 1. Wind time series is always identical for all

ice cases. The time series plots in Fig. 5 are plotted using the same averaged data used to test detection methods. The original data was drawn from the simulation at a 20 Hz sample rate. The time series in Figs. 5 and 6 consist of ten second averages. In Addition to these a big reference dataset was used as a baseline, teaching dataset and the three smaller ones were used to test different methods.

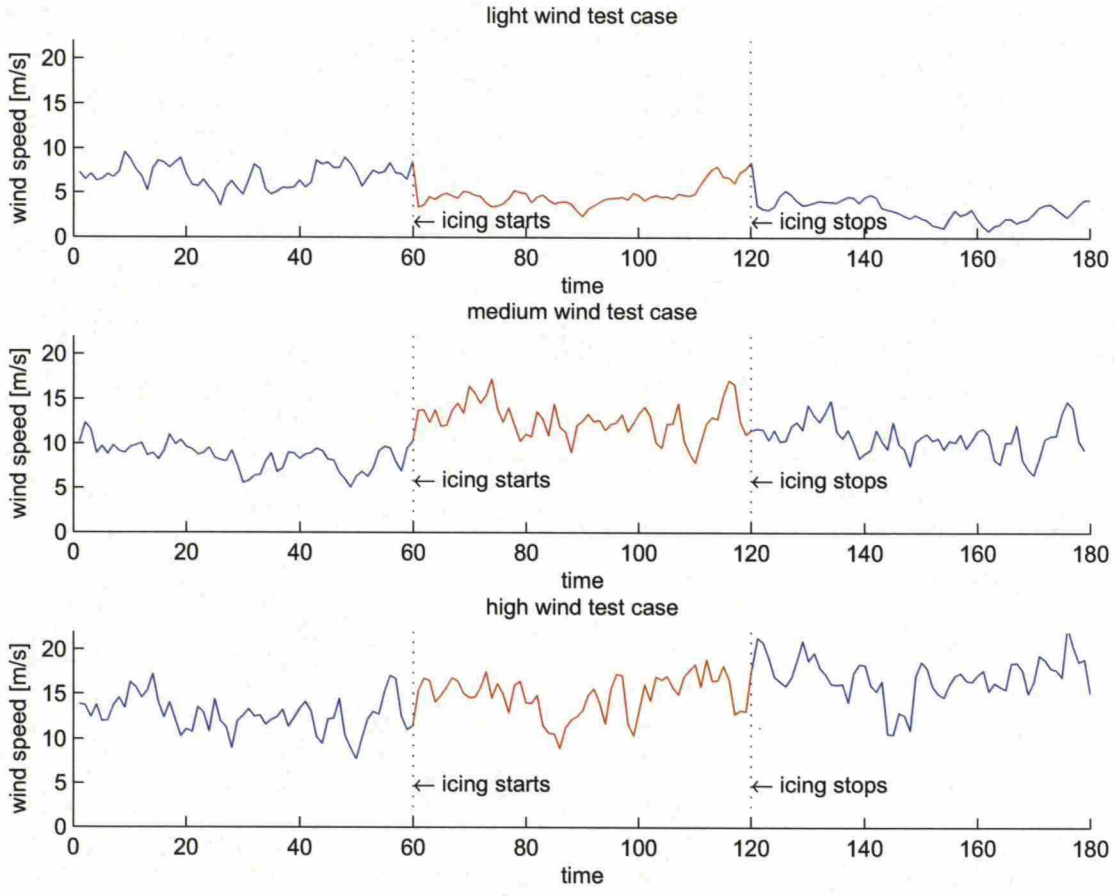

Figure 5: Time series plots of wind speed test cases

Figure 6 is plotted using the asymmetric icing cases for the simulations. The drop in power in the light wind test case in Fig. 6 at the start of icing is an unfortunate side-effect of the wind time series, it is not caused by icing. Generally, the relative drop in output power is larger in the low wind test case. In the high wind test case the turbine sometimes reaches the nominal operating power even when iced.

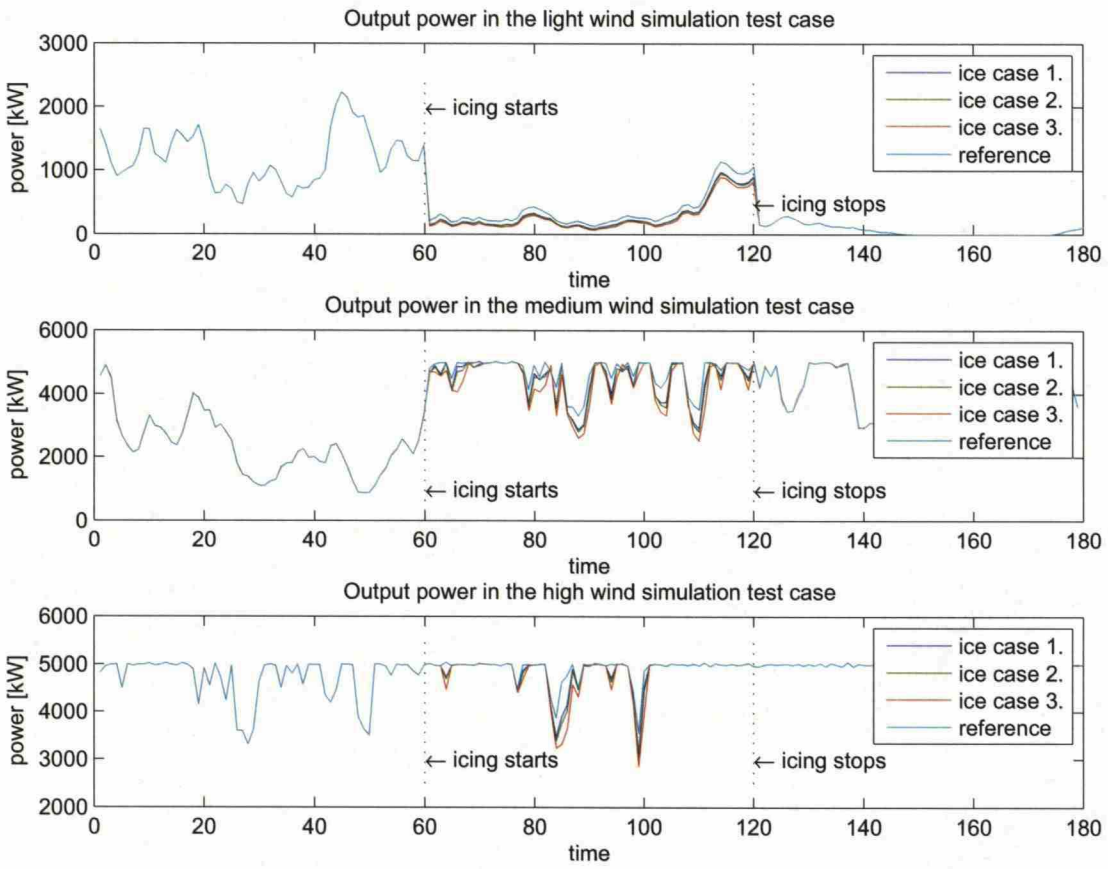

Figure 6: Time series plots of generator power in all test cases. Using ice cases from table 1

4 Indirect detection methods

The aim of this work is to create an ice-detection system that can be implemented in the future as a real-time solution. Because of this the goal is to avoid computational complexity. This aim also disqualifies methods that work by analysing large datasets recursively. Also because the goal is to create a reusable system, which is not closely coupled with an individual turbine or turbine site configuration, most interesting approaches involve the use of non-parametric models.

Ice detection can be seen as a fault detection problem. Fault detection is a well studied field with a large number of well known and tested methods. In general fault detection methods can be divided in to three sub-categories: model-based, knowledge-based and process history-based approaches. [14], [15], [16].

For the purpose of this work only statistical, process history -based models are used. This decision is mostly based on two factors: statistical models can be used without knowing the exact nature of the faults and without using an explicit model of the system. While statistical fault detection methods are often used incorporation with a model, it is not necessary, if the normal behaviour can be derived from some different source.

4.1 Non-parametric models

The idea is to create a system that can be trained to detect anomalies in the turbine behaviour. It is possible to deduct the effects icing will have on turbine by studying the simulation data or verifiable measurements. Based on this it is possible to create a system that looks for these pre-defined abnormalities. This way it is possible to also avoid fitting a complete model of the turbine inside the ice detection system. Using a model based method can be problematic, because these methods demand exact knowledge of a large number of model parameters. Wind turbine models can be constructed, but these require explicit knowledge of certain turbine characteristics that might not be available. Also using a non-parametric approach might prove to be simpler. [2],[17], [18]

The simplest possible approach would be to use several separate detectors, and a simple voting scheme to decide the final outcome. This kind of approach, where several independent detectors are used, makes developing and extending the system a very straightforward procedure. Also, using separate algorithms to detect different symptoms means that these detectors can be tuned more closely to detect a specific symptom only. Problem in this is that strict univariate methods might not work properly in a complex process due to cross correlations between process variables. On the other hand changes in correlations between variables can be interpreted as anomalous behaviour and can be detected as faults in the system even if the values of individual variables are not outside the designated alarm limits.[21],[22]

The use of non-parametric methods means that there is no need to have explicit information about the turbine. The methods operate based on process data alone. This means that the detector needs to be tuned with historical process data in order to teach it the normal behaviour of the system. A practical approach would

be to tune the detector based on data collected during warmer times, when the temperatures are high enough so that icing is extremely unlikely. As a result the detector needs to have two operating modes and we switch between those modes according to outside temperature. There are several approaches to this kind of problem and the question here is to pick the right ones that work in our case. Some examples can be found in [18], [21], [17], [23]

The selected method family does not really change the basic idea of anomaly or fault detection. There are basically two approaches that can be used to solve the icing detection problem. The first approach is to collect a larger dataset from a normally behaving turbine. Then, as the system is running, the real time measurements can be compared to the collected reference data. Any sudden changes in differences here can be interpreted as faults. The other approach is to identify a model of the observed process first. Then said model needs to be updated for each step and monitor the change in model parameters. This approach can be more difficult to implement, especially if the target process is very non-linear. [18]

Icing can be seen as an external fault in the process or, more correctly, as a sudden change in internal process dynamics. This change in dynamics happens because icing changes the aerodynamic properties of the turbine blades. Different kinds of fault detection methods can be used to detect this change. The simplest method is to define a set of variables known to be affected by icing and monitor these for abnormalities. If these process variables exhibit non-standard behaviour in favourable conditions it can be assumed that it is due to ice accretion. A big drawback in using purely non-parametric data based methods is that it might not be possible to isolate the cause of a fault. This means that false positive alerts of ice when there is no icing are likely. The number of false positives is an important performance measure for all detection methods and should be taken into account when evaluating detection method performance.[21]

It is also extremely unlikely that there is significant amounts of training data available from an actually iced wind turbine. Furthermore, it is even more unlikely to have data from different icing scenarios or data from iced turbine from all relevant wind speeds. Also there is no really accurate way to measure mass of ice on the blades, so even if there was data of icing events it is not possible to properly tell the difference between different icing events. All this means that the proper approach to this problem is to simply try to detect abnormal behaviour. This is an approach to fault detection generally known as novelty detection. The purpose in novelty detection is to compare the behaviour of the system to a known good dataset and try to determine if a measurement falls within the boundaries of this known, good behaviour. If the measurement is significantly different from the reference data, it can be assumed to be faulty. In other words novelty detection methods calculate the distance between the measurement and the reference dataset. If this distance becomes too large, the measurement is considered to originate from a faulty system.[24], [25]

It is possible to determine how certain process variables should act in the case of an icing event. But there is always a certain degree of uncertainty involved, especially, when dealing with real data. In addition to data based methods described

in this work it is possible to collect data from an ice sensor and then correlate the results for improved accuracy.

Correlating with an ice sensor is not entirely reliable way for checking whether the method produces proper results or not. The tip of the blade produces a significant part of the rotational force that spins the rotor. Because of this ice accretion near the tip of the blade has a more significant effect on turbine performance than icing closer to the root of the blade. The size of a modern wind turbine means that the ice sensor is located 50 - 70 meters away from the blade tips.

A more sophisticated way would be to treat icing as one of several possible faults in the process and look for specific fault signatures in the process data. This would require a fault "fingerprint" to be detected in the process data. Biggest benefit here would be an increased certainty that the problems seen in the data are caused by ice and not for example problems elsewhere in the turbine. This is especially relevant if these methods are used to control an anti-icing system.

There are two approaches that can be taken to process the data once a comparison dataset from non-iced data has been created. One is to collect data for a longer time, get a set of data points and calculate differences in interesting variables and differences in detected vibrations over this measurement period. The other alternative would be to read data in measurement by measurement and calculate the differences in real time. This would probably lead to faster classification, but as a method it would be a lot more sensitive to noise in measurements. Moreover a good and reliable way of dealing with false positives is required.

This uncertainty also means that it is very difficult to use these kinds of methods to detect the degree of icing. Instead it is easier to settle for a binary output signal: in the best case scenario it is possible to give a probability or an estimate of certainty. Even then, this would not be a probability for icing, it would be a probability that the turbine behaviour differs from normal. Theoretically it is possible that there is some other problem with the turbine that causes icing-like symptoms, but there is really no way to tell.

4.2 Look-up tables and variable binning

The easiest approach to tackle the non-linearities in the process is to use a set of look-up tables in place of a more complex model of the observed process. This is especially relevant because both the terrain surrounding the wind turbines and especially the wakes of other wind turbines have a significant effect on wind turbine behaviour. The easiest way to take this into account is to create a two-dimensional look-up table of the behaviour of different process variables and their behaviour as a function of wind speed. Once a look-up table is constructed, it is simple to calculate the difference between actual measurement and a reference value interpolated from the look-up table. The operating principle of this method is illustrated in Fig. 7.

The first step in constructing lookup tables is to pick a value for bin width b (usually 0.5 m/s). After this a set of bin centre points $v_b(i)$ is defined so that $v_b(k+1) - v_b(i) = b$. Measurements of process variables $x_j(i)$ are then distributed into appropriate bins according to the wind speed measured at time i. A separate

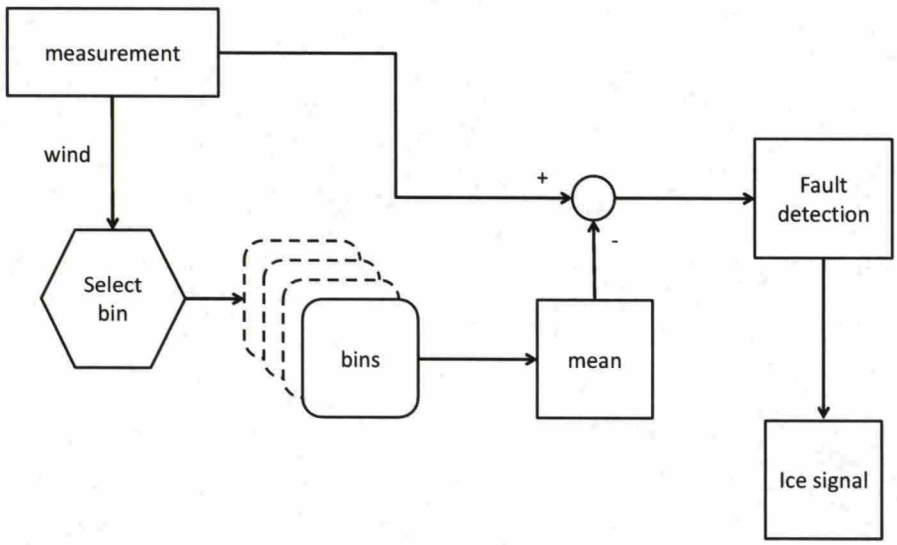

Figure 7: Using the binned data for fault detection

set of bins is constructed for each monitored process variable x_j. Figure 8 shows how generator power measurements can be fitted into bins according to wind speed measurements.

After the variables have been binned the lookup tables are made by calculating the binwise mean $m_j(i)$ of each process variable x_j separately for each bin B_k according to (3). The final lookup-table consists of bin centre points and these binwise means. For other purposes we also create a table that contains the bin sample standard deviation $s_j(i)$ for each variable with (4). In (3) and (4) n_k is equal to number of variables in bin k.

$$m_j(i) = \frac{1}{n_k} \sum^{B_k} x_j \tag{3}$$

$$s_j(i) = \frac{1}{n_k} \sum^{B_k} (x_j(t) - {m_j}^2) \tag{4}$$

When a measurement is read at time i, the appropriate bin is selected according to wind speed $v(t)$ by picking the bin center point $v_b(i)$ closest to the measured wind speed $v(t)$. Bin index k is saved. Now that the bin index is known, the values for other process variables can be interpolated from the lookup table. The result is a comparable set of data points that correspond to the wind speed measurement. The only information that needs to be saved about the bins is the coordinates of the bin center point. This requires an extra step when interpolating: if the measurement

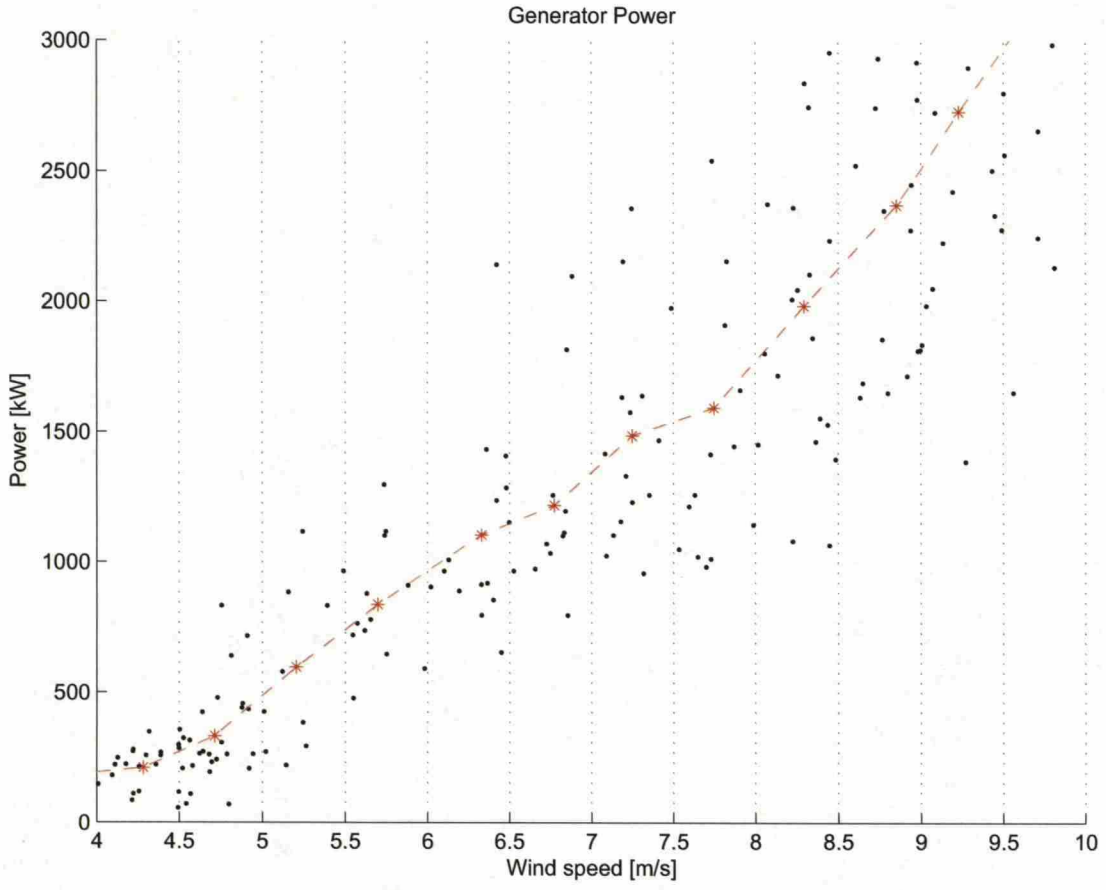

Figure 8: Simulated values of turbine power placed into appropriate bins, and the binwise mean plotted on top

belongs to bin B_k but is larger than bin centre point $v_b(i)$, the corresponding reference value $\hat{x_j}(t)$ is interpolated with (5). If the value is in bin B_k, but smaller than bin centre $v_b(i)$ (6) is used.

$$\hat{x_j}(t) = \frac{m_j(i-1) - m_j(i)}{v_b(i-1) - v_b(i)}(v(t) - v_b(i)) + m_j(i) \quad (5)$$

$$\hat{x_j}(t) = \frac{m_j(i) - m_j(i+1)}{v_b(i) - v_b(i+1)}(v(t) - v_b(i+1)) + m_j(i+1) \quad (6)$$

In (5) and (6) $v_b(i)$ refers to average windspeed in bin k and $m_j(i)$ refers to average value of the process variable in bin k. $\hat{x_j}(t)$ is the corresponding interpolated reference value at time i.

Using look-up tables makes it possible to capture real turbine behaviour and its inherit non-linearities as simply as possible. The relationships between wind speed and different process variables are non-linear and not always easily modelled. Using look-up tables is also computationally very efficient. This is very beneficial when attempting to implement these methods in practice. These look-up tables can also

be repopulated when conditions are safe enough so that icing events are unlikely to happen. It is possible to collect data and update the look-up tables at times when temperature is safely above zero and switch to icing detection mode when the temperature drops to a lower level. Real-time updating makes it possible to "seed" the look-up table manually with predefined values e.g. the turbine power curve and expected values for rotational speed and blade pitch angle. These predefined values can then be later updated with real-time measurements.

In this work the most used approach is to use look-up table as a filter. This makes it possible to take external error sources such as neighbouring turbines into account and standardise the data first before trying to detect any strange behaviour.

Bin size does not have to be constant though, the only requirement is that the variable behaviour does not change too much inside the bin, because the interpolation requires the assumption that variable behaviour is linear inside each bin. For example, using wider bins for generator power at higher wind speeds is possible, because after the turbine reaches its nominal power, it stays there until wind speed reaches the cut-off speed. This means that power and rotational speed remain close to constant for a wide range of wind speeds. This can be taken into account when designing the look-up tables.

Once the data has been distributed into the appropriate bins, the next step is to calculate the bin sample mean for the process variable and use these means and bin centre points to create the final look-up table. The actual measurement data can contain outliers that can affect the mean, so in some cases it might be more useful to use a more robust estimate for the mean. Different approaches here could be to use the sample median or just simply reject the measurements that differ from the mean most. For example, if the top and bottom 5 % of the measurements are removed from the data, the amount of outliers will be significantly smaller. [20]

It is also very straightforward to keep these look-up tables up to date. The only real requirement here is to discard old data as new data becomes available. This is done by setting an upper limit N for the bin size updating the bin by following a first-in-first-out approach. When the number of measurements in a bin reaches N the oldest measurement is removed and a new one is added to the bin. The update method is illustrated in Fig. 9.

Once the look-up tables have been constructed they can be used as a crude approximation of the turbine behaviour, as a replacement for a proper model. This makes it possible to investigate the turbine behaviour as it relates to wind speed and makes it easy to normalise the measurements so that differences in different process variables can be compared with one another. The most direct method for variable normalisation is to divide the difference between the interpolated value from (5) or (6) and the current measurement x_i with the bin standard deviation according to (7).

$$z_j(t) = \frac{x_j(t) - \hat{x}_j(t)}{s_k} \tag{7}$$

Normalisation like this eliminates most of the effects that widely different absolute values of process variables have in some detection methods. This does double

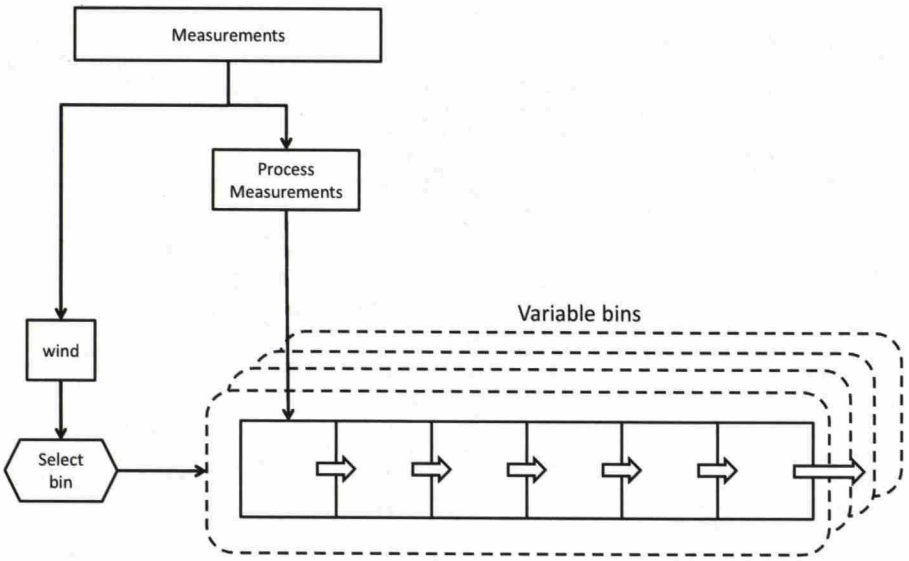

Figure 9: Basic update procedure of a single bin. New measurement is added and oldest is dropped.

the amount of data that needs to be stored, but it improves the method accuracy significantly, so trade-off can be considered worthwhile. In cases where the bin standard deviation is zero the normalised value is set to zero.

Normalising the data improves the detection accuracy, especially when using several concurrent measurements. Normalisation requires a separate set of look-up tables for variable standard deviation. These can be calculated binwise the same way as tables for the sample mean are done. And just like with mean here outliers can affect the results to some degree. When using real data a more sophisticated method for outlier elimination might be needed.

A real concern here is how a too small number of measurements in a single bin affects the outcome of the normalisation. Too small number of measurements can cause the deviation within the bin to rise, especially with noisy data. This is problematic, because the normalisation will produce odd results this way. An example is illustrated in Fig. 10 for sample standard deviation.

The curves in Fig. 10 are calculated from real, somewhat noisy data, that has some outliers. A similar effect, but in opposite direction can be seen for mean as well. This illustrates the need for a good method for outlier removal. On the other hand, the sample size needs to be reasonably large in order to be able to classify a value as an outlier in the first place. A quick solution for this could be to set a lower limit for bin size. A low limit for bin size would mean that bins are only activated once there is enough data in them. Alternative approach would be to seed the bins with data from simulations or measurements from other turbines of

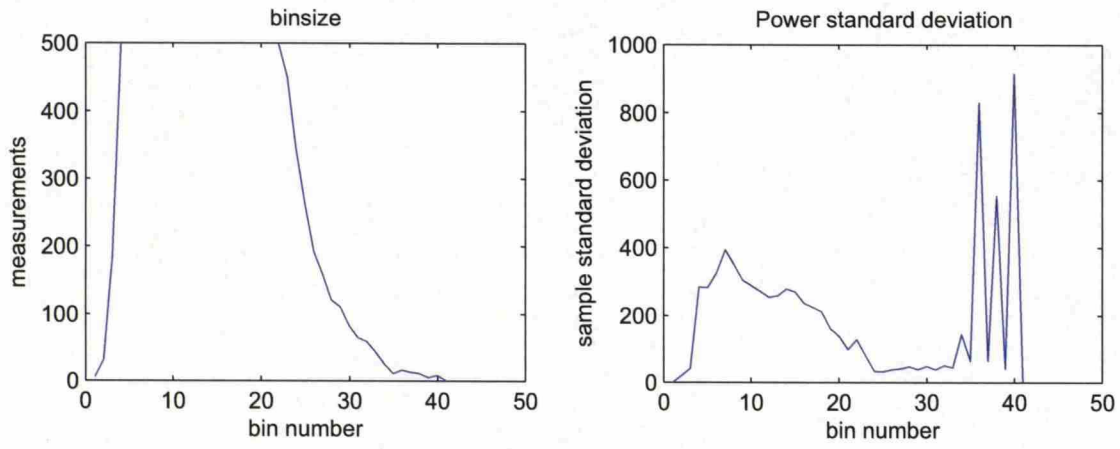

Figure 10: Effect of bin size on bin sample standard deviation

the same type. This data can then be updated with measurements from the actual turbine. Pre-seeding the lookup tables would eliminate the requirement to collect data first before the ice detection system can be properly activated.

Changes in air temperature can affect the measurements. This change can have a noticeable effect on detection accuracy, because the basic principle here is to use data collected at summertime to gauge turbine performance during wintertime. The main culprit here is the fact that air density is inversely proportional to air temperature and directly proportional to power produced by the turbine at a set wind speed.

Air density effects the amount of power a turbine produces according to formula (2). This might have an effect on ice detection, because the power curve will shift to the left on colder weather. As a result the turbine produces slightly more power at the same wind speed on colder weather. This has a direct effect on efforts to detect icing from process data. When comparing the raw measurements to a power curve defined at different conditions, smaller changes in process variables are not found, because the relationship between wind speed and the process variables has changed slightly.

According to [26] the measurements can be normalised for changes in air density by normalising the wind speed with with (8). In (8) $v(t)$ is the measured ten minute average of the wind speed, p is the ten minute average of current air pressure, T is the ten minute average of the temperature and R_0 is the gas constant of dry air ($R_0 = 287.05\ J/(kg * K)$), ρ is the reference density of air.

$$v_n(t) = v(t) \left(\frac{p}{RT\rho_0} \right)^{\frac{1}{3}} \tag{8}$$

Figure 11 shows the effect of temperature correction on real data. Note that there were no pressure measurements for the data set used, so it is assumed that the pressure stays constant at $p = 1000\ hPa$ for all measurements. Still the effect of

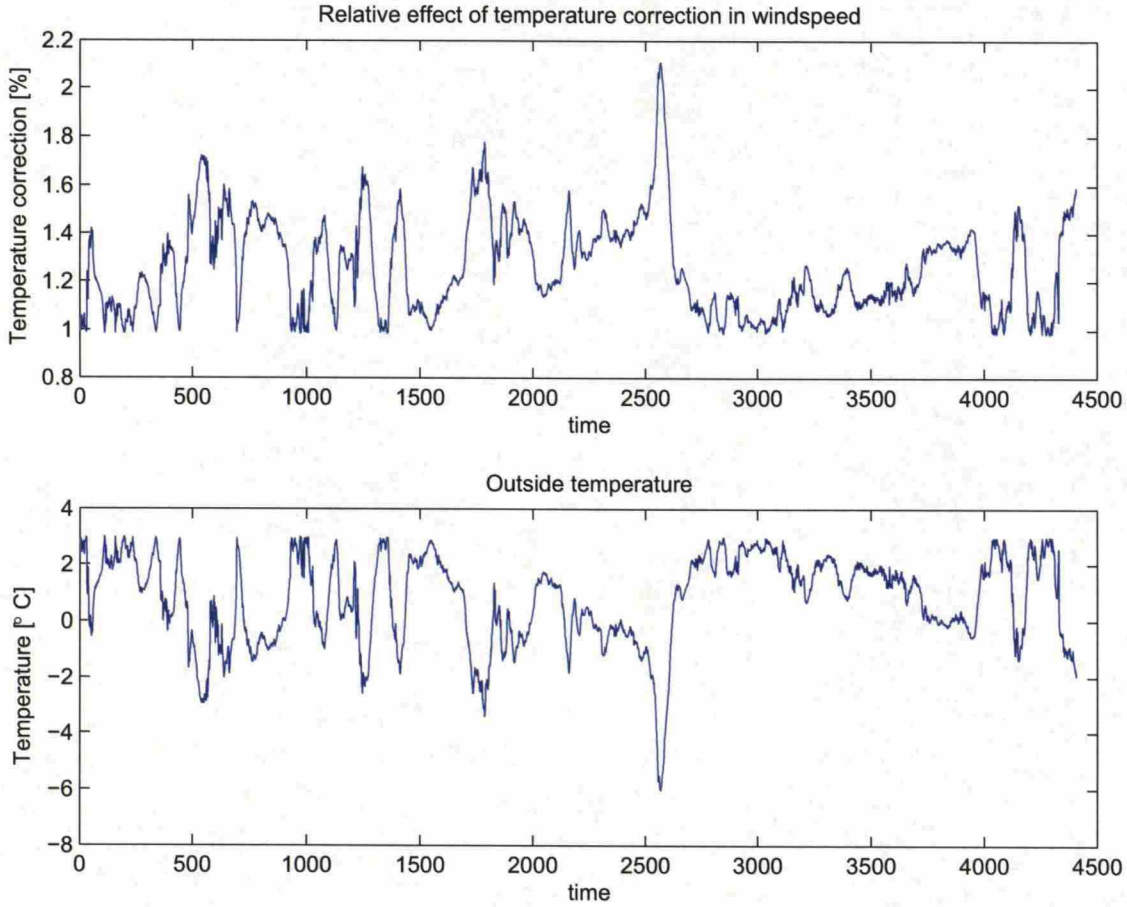

Figure 11: Relative size of temperature correction compared to measured wind speed

temperature alone is generally noticeable. In the measured dataset it is equivalent of a 2% increase in wind speed.

The normalized wind speed can then be used as input for normal interpolation according to (5) or (6) in order to get the reference values for all measurements.

4.3 Control charts

The most straightforward solution to fault detection would be to create a control chart that acts as an indicator of the state of the turbine. Control charts are an important concept in statistical process control and are often used as a tool in automatic quality control and fault detection. Control charts are used to determine whether a process is in control or not. A process is considered to stay in control if nothing unusual (e.g. a process fault) happens. An in-control process behaves in a predictable way. Anomalies from normal behaviour are interpreted as a sign of an out of control process and can be interpreted as signs of a process fault. Control charts are used to track values of a monitoring variable, that acts as an indicator of

process state. Monitoring value is often a specifically calculated statistical quantity that acts as an indicator of the process status.

When a process stays in control, the value of the control chart variable stays at a predictable level. Faults in the process cause the value of the control chart variable to grow. Too large values can then be interpreted as signs of a fault or an out of control process. For the purposes of ice detection, it is possible to create a control chart using the variables that are affected by icing to detect changes in them, and to create a single signal that can act as an indicator of icing events.

Most common control charts are so called CUSUM (cumulative sum) and EWMA (exponentially weighted moving average) charts. The idea is to track the values of interesting process variable and monitor its deviations from the long term mean. If the values of the observed variable deviate too much from mean the process can be deemed to out of control i.e. exhibiting abnormal behaviour. Both CUSUM and EWMA are essentially univariate charts meant for a simple SISO process. These univariate methods are easy to implement and have been shown to be reliable, but can sometimes fail when used with a process with multiple correlated variables.[21],[23]

Standard CUSUM chart means plotting the cumulative sum defined in (9). Here x_j is the monitored process variable and μ_0 is a reference value for the process mean. The cumulative sum $S(t)$ starts to grow when $x_j(t)$ starts to deviate from the mean. Process is considered to be out of control when the value of $S(t)$ grows too large.[27]

$$S(t) = \sum_{i=0}^{t} (x_j(i) - \mu_0) \tag{9}$$

To properly detect changes in the value of the sum we need to define limits for S_t. Here the so called decision interval scheme is used to define the control limits for the cumulative sums. In order to be able to compare changes in different process variables to each other, all the variables need to be scaled accordingly. Most effective approach here is to normalize the variables first according to (10). Here $\overline{x}$ is the sample mean of variable x and σ is the sample standard deviation. [27]

$$z_j(t) = \frac{x_j(t) - \overline{x}}{\sigma} \tag{10}$$

Normalising the variables this way produces a set of residuals that are more comparable to each other. And more importantly it is a set of variables where it is easy to define appropriate limits to determine if the process is out of control. In order to properly detect out of control behaviour of the normalized variable two variables, S_{Hj} and S_{Lj} defined in (11) and (12) need to be calculated. The process is considered to be out of control if $S_{Hj}(t) > h$ or $S_{Lj}(t) > h$. Here k and h are constants that need to be set on a case by case basis. k is called a reference value and h is known as decision interval. [27]

$$S_{Hj}(t) = max(S_{Hj}(t-1) + z_j(t-1) - k, 0) \tag{11}$$

$$S_{Lj}(t) = max(S_{Lj}(t-1) - z_j(t-1) - k, 0) \tag{12}$$

For normalised variables z_j k can be seen as the size of the anomaly we want to detect. The decision interval h can be interpreted to be related to probability of false alarms. Both of these parameters need to be adjusted individually for the process at hand. [27]

For normalised variables it can be assumed that they behave in somewhat similar manner so that the same parameters can be used for each of variable.

For a multivariate process a different approach that takes the different correlations between the process variables into account is needed. There are several multivariate statistical process control (MSPCA) methods that can be used to monitor a more complex process. Most of these are simply extensions of the univariate methods and are relatively easy to implement. Alternative approach would be to calculate a set of univariate control charts and then combine the results. [21][17][23]

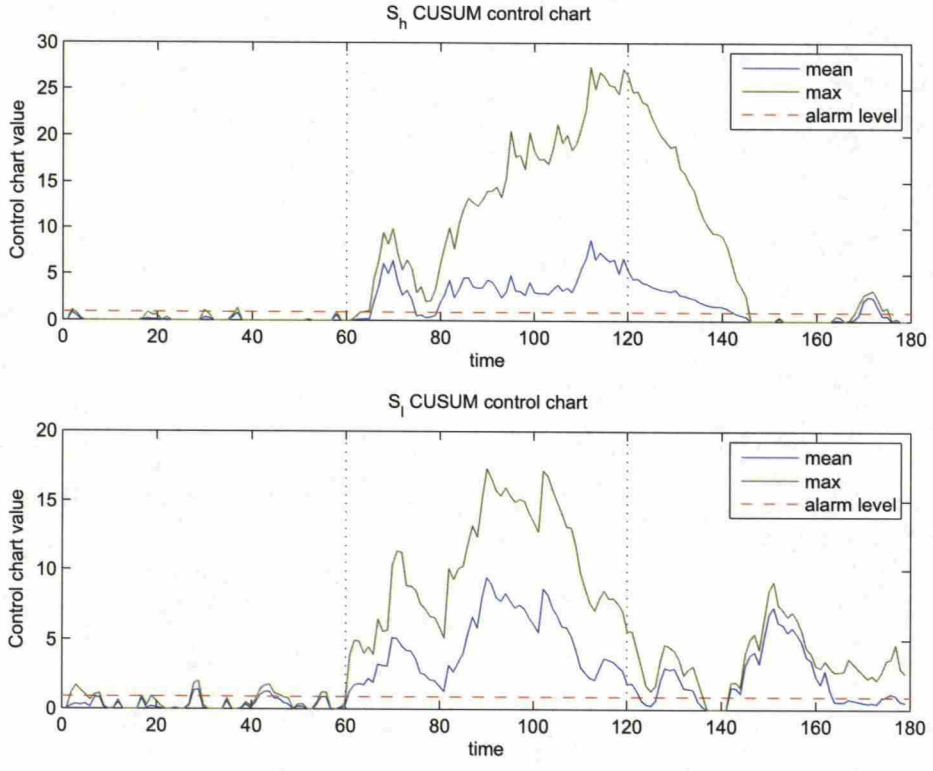

Figure 12: Effect of the chosen method to create a compound chart

In order to create a combined chart, a simple solution would be to combine the values of individual charts for different process variables. Because normalised variables are used instead of raw, unscaled measurements, the values of all control charts are comparable with one another. As a result two alternative approaches to chart combining problem can be taken. Either combine the charts and use one upper limit and compare against that, or specify limits separately for each chart

and combine the results.

If the goal is to detect an anomaly in process behaviour as quickly as possible an alternative approach would be to use the maximum of all of these values. This is lot less robust approach and it will increase false positive results, but it will react faster. One approach to make it a bit more robust is to demand that more than one chart needs to cross the alarm line before an alarm is raised. This should make the system a bit more robust to fluctuations in individual measurements.[28]

Table 2: Effectiveness of different compound chart construction methods

	correctly detected iced [%]	correctly detected clean [%]
S_h Max	93	68
S_h Mean	82	77
S_l Max	98	76
S_l Mean	58	89

Figure 12 illustrates the effect of the chosen compound chart construction method. Using mean will produce fewer false alarms on the simulated dataset, but the signal level is overall lower. As a result the S_h chart level drops below the warning level before the icing event has ended. The results are collected in Table 2 for one simulation case. Table 2 lists the percentage of correctly detected iced measurements and percentage of correctly detected clean (not iced) data points. From these results it is clearly visible that using maximum values increases the detection accuracy and increases the number of false positive detections. In the case used to build Table 2, the drop in detection accuracy is a lot more severe than the drop in correctly detected clean data points. On basis of this using maximum values would be a better option at least for the S_l-chart.

In addition to this, these charts will only detect changes in one direction: values of S_h-chart will grow if the value of the test variable grows and value of S_l-chart will grow if the value of the test variable becomes smaller. The final alarm should then be issued when either of the two charts crosses the alarm limit because these two charts are complementary. There are no process variables that contribute to values of both of these charts at the same time, because the changes caused by icing will be to same direction i.e. certain variables will always see a drop in their value and certain variables will see their value grow as a result of icing.

Different, a more direct approach is to use a specifically defined multivariate chart. This has the added benefit of also taking the correlations between the variables of the system in to account. In a complex process like a wind turbine the process variables are always interconnected to a degree. Combining univariate charts will always carry the assumption that the variables are independent to a degree. In some cases this will have an effect on detection accuracy. The most common alternatives are multivariate CUSUM and multivariate EWMA charts. These are simply multivariate versions of common univariate charts. A multivariate EWMA chart is defined according to (13).[30] [31]

$$\mathbf{z(t) = Rx(t) + (I - R)z(t - 1)} \tag{13}$$

Here $\mathbf{R} = diag(r_1, r_2, r_3, ...)$ and $0 \leq r_k \leq 1$, and x_i is the monitored variable. It is assumed that $x_i \sim N(\mu_0, \Sigma_0)$ with a known covariance matrix Σ_0 and known mean vector μ_0. The chart gives an out-of-control signal if $Q_E > h_E$ in eq. (14). There isn't a well specified way to determine the control limit h_E, it needs to be evaluated from data collected from a known safe case or calculated from simulation data using e.g. Monte Carlo methods. [29]

$$Q_E = z(t)^t {\Sigma_{z(t)}}^{-1} z(t) \tag{14}$$

Assuming all variables in the input data are equally relevant, it is possible to set $r_1 = r_2 = r_3 = ... = r$. In this case the covariance matrix ${\Sigma_{z(t)}}^{-1}$ is defined by (15).

$$\mathbf{\Sigma_{z(t)}} = \frac{(1 - (1 - r)^{2t})r}{2 - r}\mathbf{\Sigma} \tag{15}$$

This of course requires exact knowledge of the process variable covariance matrix $\mathbf{\Sigma}$.[29] The method is in general rather sensitive regarding the initial covariance matrix. Detection sensitivity is greatly affected depending on the data that was used to define the covariance matrix. The best solution is to use a properly normalised portion of the reference dataset. Detection works with simulated data also when using a covariance matrix calculated from the iced data set, and even better when using just a safe portion of the iced data set. This option is not as practical, because a full iced data set is not available when attempting to detect icing events in real time.[30]

EWMA control charts can be constructed also for non-normalised data. The detection does work in these cases, and for simulated data, using raw, non-normalised variables in control chart creation produces larger changes in the final signal when icing is detected. Normalising the data makes the method a lot less sensitive to the definitions of the covariance matrix and the parameter r.

Some kind of safe upper limit for the control charts is needed to determine whether the process is in an out of control state. Upper limits for multivariate EWMA control charts would need to be calculated from data and there are often no direct methods for the multivariable case. Often the suggested method is to use simulations to gauge an estimate for the out-of control alarm limit. The basic process here is to simply run a lot of simulations and decide an appropriate level for the chart based on these results. [32]

A less formally correct approach here would be to calculate the values of the control chart using the reference dataset as input. From there we can calculate an assumption for the upper limit for the control chart variable. An example is illustrated in Fig. 13. A value of 120 % of the maximum of the control chart value from the reference dataset is used to define the alarm limit. The dashed red line is the alarm limit. In Fig. 13 the figure on the left is generated by calculating the

values of the control chart using the reference data as the input. The graph on the right shows the control chart value over time for a sample icing incident. The alarm limit stays the same in both cases.

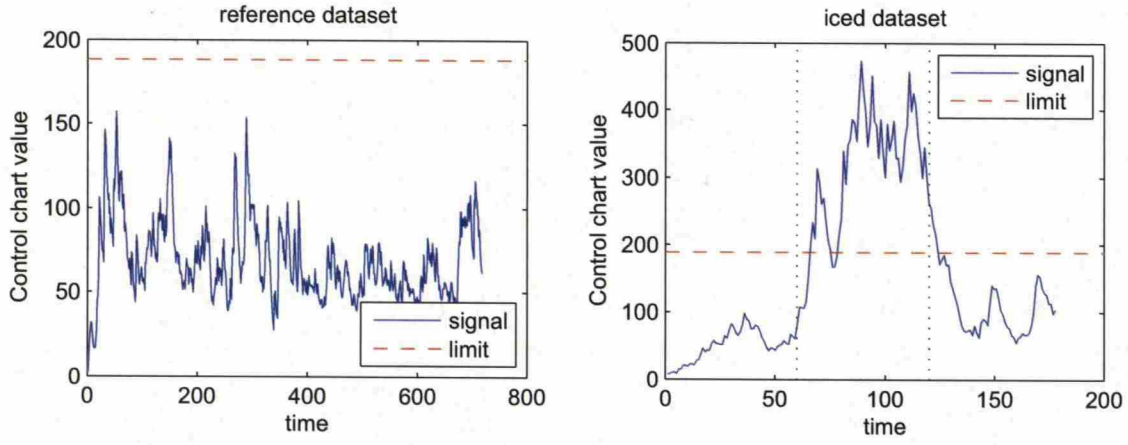

Figure 13: Setting the alarm limit to 120 % of the maximum of the control chart value in the reference dataset. The beginning and the end of the icing incident are marked with a vertical dotted line on the graph on the right.

Here the interesting feature is the relative change in the signal amplitude. When looking for anomalies in the process, the bigger the change from a base level, the easier it is to isolate the change as relevant. When using EWMA charts the only adjustable parameter is the smoothing parameter r in (13) and (15). Safest assumption is to assume that $r_1 = r_2 = r_3 = ... = r$ meaning that all variables behave in similar fashion over time. Adjusting the value of r adjust the speed at which new information is brought in to the chart. Suitable value of r depends on the number of handled variables and also on the magnitude of the shift in variables caused by the fault. [30] [29]

The choice of r also affects the speed and sensitivity of the detection. Figure 14 shows the effect of changes in r on a MEWMA control chart plotted for a simulated dataset. There is an icing event in the dataset on the interval [60,120]. It is easy to see from Fig. 14 that too small values for r make detection slower and also make the signal fall back to normal level slower. On the other hand larger values for r result in smaller absolute values for the signal. The simulation data is cleaner than real data, which results in easier detection and makes it possible to remove false positives completely. When using real measurements too large values for r can cause the number of false positive detection signals to increase because of noise in the data.

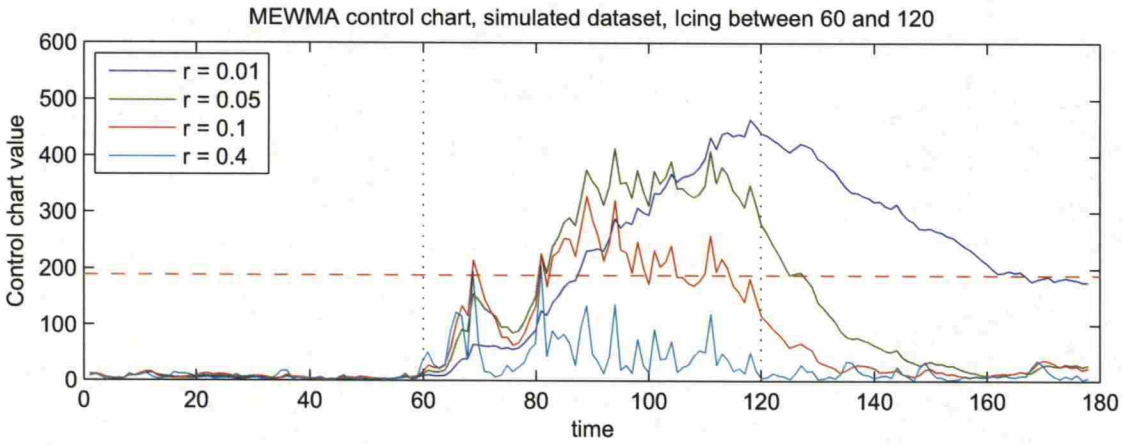

Figure 14: Effect of parameter r on multivariate EMWA chart. Alarm limit marked with the dashed red line.

It is also possible to create a multivariate version of a standard CUSUM-chart. One approach is to simply replace the standardised individual variables with the D^2 statistic defined in (16). D^2 can be seen as a multivariate version of squared error. Strictly it is the square of the Mahalanobis distance between he measurement and the mean. It is similar measure than the normal distance, except that it also takes the correlations between different variables into account.[32]

$$D^2(t) = (\mathbf{x(t)} - \mu)^{\mathbf{T}} \mathbf{\Sigma^{-1}} (\mathbf{x(t)} - \mu) \tag{16}$$

Calculating the D^2 statistic as defined in (16) once again requires exact knowledge of the covariance matrix. A possible approach is to use a covariance matrix calculated from the reference dataset. There are multiple different methods to construct a multivariate CUSUM chart, some of which have bee collected in [32]. The method we use here is very similar to the univariate methods described by eq. (11) and (12). The difference here is that the D^2 statistic can only have positive values so we need to calculate only one sum. [32]

$$S_M(t) = max[(S_M(t-1) + D(t) - K), 0] \tag{17}$$

Here $S_M(0) > 0$ and $K > 0$. The process is deemed to be out of control when $S_M(t) > H$. Where H is a predetermined constant. K is related to the magnitude of shift in process variables that requires an alarm to be raised.

Using a set of individual single variable charts as opposed to actual multivariate methods can be justified, if similar performance can be achieved while avoiding computational complexity that might come with multivariate methods. Mathematically simpler methods can sometimes be preferable. It is possible that these methods need to be implemented on a system that cannot do matrix algebra. Inverting the

covariance matrix can also in some cases lead to numerical problems because the process variables are correlated.

An additional prefiltering trick that can be used to improve the performance of our control charts is to take the sign of the change into account. It is known what kind of effect icing will have on at least some of the measured process variables. The magnitude of the change might not be known. Even less likely is that the exact relationship between the change in process variable and the amount of ice on the turbine blade is known. But in most cases it is possible to tell the direction of the change icing causes in the process variable.

The direction of the change in process variable values induced by icing can be deduced from e.g. simulation studies or from prior measurements. Even if the magnitude of the change in process variable values is not known, knowing the sign of the residuals in an icing event is enough to help create a simple filtering scheme that has noticeable effect on detection accuracy. The number of false positives can be reduced by filtering wrong signs away form the data. For a given measured process variable x_i there is a corresponding reference value $\hat{x}_j$. Most of these methods deal with differences $d_i = x_i - \hat{x}_j$ between these two. Now if the direction of change that icing causes in a variable (e.g. drop in power)is known, the variable can be filtered according to (18), and conversely for variables that are known to grow (loads) (19) can be used.

$$d_{x,j}(t) = min(0, x_j(t) - \hat{x}_j(t)) \tag{18}$$

$$d_{x,j}(t) = max(0, x_j(t) - \hat{x}_j(t)) \tag{19}$$

Most methods discussed here do not differentiate based on the direction of change in variables. By removing changes in unwanted directions it is possible to get rid of at least one potential source of false positive detections. Curiously doing a sign based removal of features makes detection performance worse on simulated datasets. The relative change of control chart variable is smaller when calculated this way. On real data this has proven to be a useful trick to reduce false positive detections.

4.4 Principal component analysis and fault detection

In some cases when dealing with very large-dimensional data, some of these variables may be highly correlated with one another. In these cases it would be beneficial if it were possible to reduce the data dimension without losing too much valid information. One way to achieve this is to use principal component methods. Principal component analysis (PCA) is an often used method to transform the original correlated data into a smaller set of uncorrelated data while preserving as much information from the original data as possible.

Another benefit of using PCA methods is that it is not required to explicitly choose the variables used for detection, the method finds the most meaningful variables automatically. Meaningful here means the variables that best describe the changes in the reference dataset, not the variables that are most affected by icing.

The reduced small set of vectors can be used as a model for the full wind turbine system. Here it is important to note that PCA is a linear method, and the observed system (wind turbine) is somewhat non-linear. The problems caused by nonlinearity can be alleviated to some degree by normalising the data first as described earlier.

Assume an $n \times m$ matrix $\mathbf{X}$, that contains m measurements of n process variables $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}, ..., \mathbf{x_n}$. These variable might be correlated with each other. PCA transforms these n correlated variables into a uncorrelated variables, where $a << n$.

The first step is to rescale the datamatrix $\mathbf{X}$ into a new matrix $\mathbf{Z}$ with zero mean and unit variance. This means that each column vector $\mathbf{z_i}$ of $\mathbf{Z}$ has a mean of 0 and variance of 1, (20). Then the covariance matrix $\mathbf{R}$ of $\mathbf{X}$ is calculated as shown in (21).[33][34]

$$\mathbf{z_i} = \frac{\mathbf{x_i} - \overline{x_i}}{\sigma} \tag{20}$$

$$\mathbf{R} = \frac{1}{n-1}\mathbf{Z^T Z} \tag{21}$$

Principal components of $\mathbf{X}$ are the eigenvectors $\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}, ..., \mathbf{p_m}$ of $\mathbf{R}$. The eigenvalues $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_m$ tell the amount of variance of original data has along each principal component axis. After the principal components have been calculated an m x a projection matrix $\mathbf{P}$ can be defined by choosing a most significant principle components. Once the projection matrix has been defined it is possible to calculate so called principle component scores $\mathbf{T}$ with simple projection according to (22). Finally a projection back to the original data space can be defined according to (23). The residual (estimation error) matrix $\mathbf{E}$ is then defined as per (24).[34][35]

$$\mathbf{T} = \mathbf{XP} \tag{22}$$

$$\hat{\mathbf{X}} = \mathbf{TP^T} \tag{23}$$

$$\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}} \tag{24}$$

The proper number of principle components can be calculated from the eigenvalues λ_i using different methods. The most straightforward selection method is plotting the eigenvalues, sorted largest to smallest, and looking for the "knee" in the graph, meaning the point where the values become smaller compared to the first few eigenvalues (4 in Fig. 15). Figure 15 is calculated from the same simulation data used to generate the results in Chapter 5. Another option is to simply pick a threshold h_a and the solve a from (25). This is known as the accumulative contribution rate (ACR). ACR tells us the percentage of total variance the selected a principal components explain[33].

$$h_a = \frac{\sum_{i=0}^{a} \lambda_i}{\sum_{i=0}^{n} \lambda_i} \tag{25}$$

The principal component representation of the process can be used as a model for fault detection. The basic principle is simple: a new measurement for each process

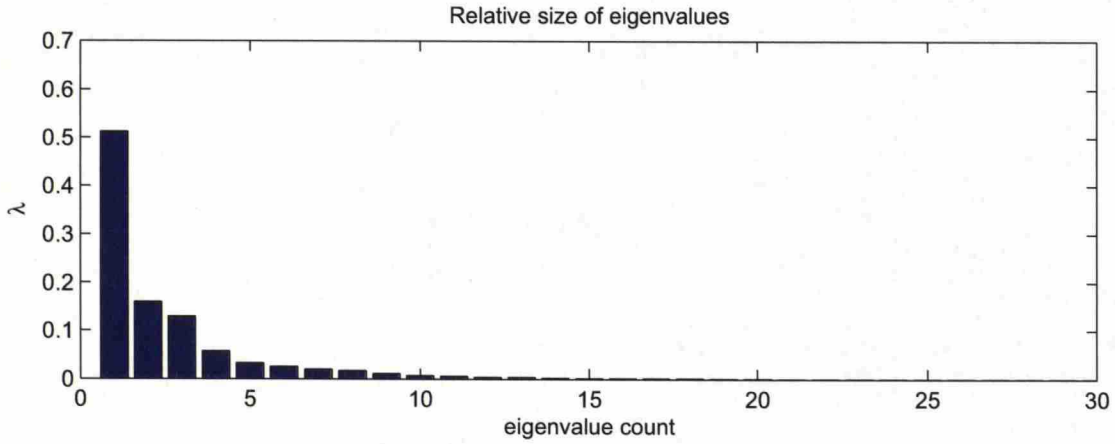

Figure 15: The eigenvalues of covariance matrix X.

variable is collected and placed in a vector $\mathbf{x}(t) = x_1(t), x_2(t), x_3(t), ..., x_m(t)$. Then the scores are calculated using the principal components defined earlier according to (26). These scores are then used to calculate a projection back to original dimension. Finally the residual r is calculated with (28).[35]

$$\tau_{\mathbf{j}}(t) = \mathbf{x_j}(t)\mathbf{P} \tag{26}$$

$$\hat{\mathbf{x}}_{\mathbf{j}}(t) = \tau_{\mathbf{j}}(t)P^T \tag{27}$$

$$\mathbf{r}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t) \tag{28}$$

These variables can be used for fault detection by constructing a control chart, that will tell us whether the process remains in control or not. The most common control chart to use in these situations are Hotellings T^2 and Q statistic. Both these can be updated for each measurement in real time, and it is possible to define control limits for both of them. For added accuracy it is recommended to use both charts side by side. This is recommendable because T^2 and Q charts are complementary. In other words T^2 chart monitors changes in variance explained by the first a principal components. The Q statistic on the other hand monitors changes in variance of the residuals.[35][34][17]

The T^2 statistic is calculated with (29). Here t_i is the score representing the principal component i and λ_i is the corresponding eigenvalue of the original correlation matrix. Confidence limits of the T^2 statistic can be then shown to be related to the F-distribution according to (30). [17]

$$T^2 = \sum_{i=0}^{a} \frac{t_i^2}{\lambda_i} \tag{29}$$

$$T^2_{m,p} = \frac{(m-1)p}{m-p} F_{p,m-p} \tag{30}$$

In (30) $F_{p,m-p}$ is the value at the upper tail of the F-distribution at confidence level α with p and $p-m$ degrees of freedom. This allows us to calculate an alarm limit for the T^2 statistic. Now if the value of T^2 exceeds the value of $T^2_{m,p}$ it means that a change in the part of the process represented by the first a principal components has been detected with a probability of α. The T^2 statistic is calculated based on the assumption that the principal component scores follow a multinormal Gaussian distribution [19]

The T^2 statistic is not always enough by itself to detect changes in the process dynamics. If there is a new event that changes the way a process behaves it might be easier to detect by monitoring the residuals. This requires us to define a complementary control chart, often called the Q chart. The Q chart is defined according to (31), where r is the residual from (28).[19]

$$Q_i = r_i r_i^T \tag{31}$$

Confidence intervals for the Q statistic can be defined according to Eq. (32), (33) and (34). Equation 32 calculates an upper limit which means that if $Q > Q_\alpha$, the process is in an out-of-control state and there is something out of the ordinary going on. Here c_α is the $1-\alpha$ confidence interval from the standard normal distribution. Use of the standard normal distribution here means that it is assumed that residuals (model error) follow a normal distribution. h_0 and Θ_i are helper variables that are used to make the formula 32 easier to read. [17]

$$Q_\alpha = \Theta_1 \left(1 + c_\alpha h_0 \frac{\sqrt{2\Theta_2}}{\Theta_1} + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1^2} \right)^{\frac{1}{h_0}} \tag{32}$$

$$\Theta_i = \sum_{j=a+1}^{n} \lambda_j^i, \qquad i = 1, 2, 3 \tag{33}$$

$$h_0 = \frac{1 - 2\Theta_1\Theta_3}{3\Theta_2^2} \tag{34}$$

The result here is two separate test variable plots illustrated in Fig. 16. Figures are drawn with a simulated dataset, where the turbine is iced between time instances 60 and 120. Figure 16 contains the two test variables and their 99 % confidence levels. We can then use a simple logical operation as a final icing signal here. Either we can issue a warning when both signals indicate abnormal behaviour or just with either one.

Normally principal component methods are defined for static data. In order to capture the behaviour of a dynamic system the data original matrix $\mathbf{X}$ is amended by adding in a set of delayed variables. This means going from $\mathbf{X} = [\mathbf{x}_1(k), \mathbf{x}_2(k), ..., \mathbf{x}_n(k)]$ to $\mathbf{X}_a = [\mathbf{x}_1(k), \mathbf{x}_1(k-1), \mathbf{x}_2(k), \mathbf{x}_2(k-1), ..., \mathbf{x}_n(k), \mathbf{x}_n(k-1)]$. Adding delayed measurements into the data matrix doubles the data size and as a result requires that a

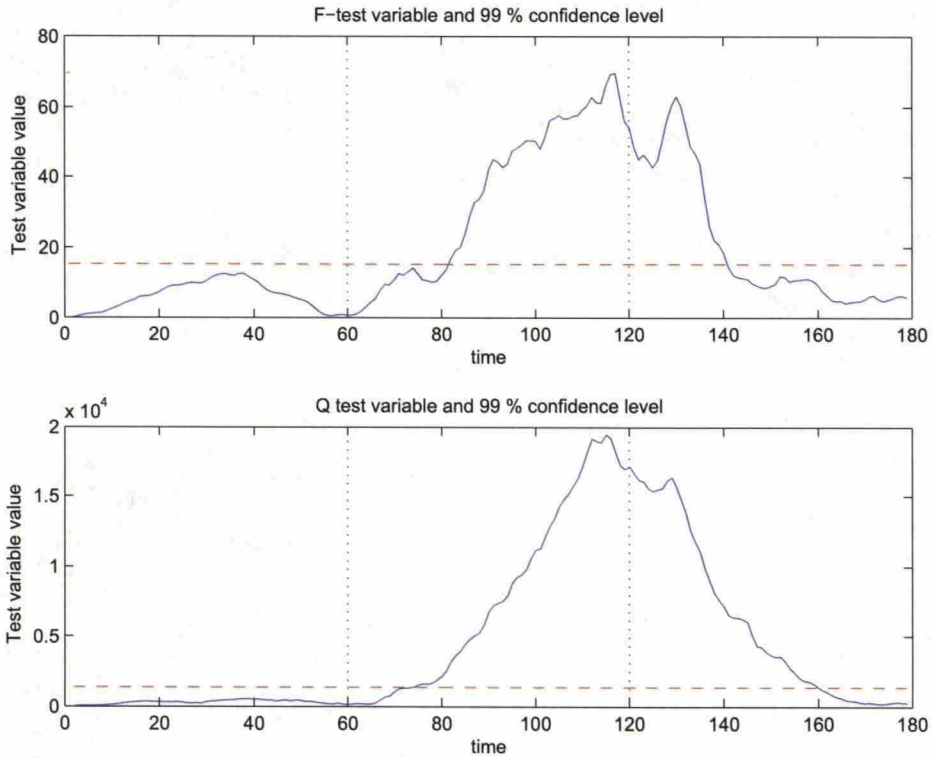

Figure 16: The PCA test variable charts.

larger number principal components used for detection, but it improves the accuracy of the used methods, especially when dealing with faster data. [18]

Time shifting can cause problems if the time delays in the process are not multiples of the sampling interval [22]. When dealing with icing it can be assumed that the icing process is so much slower than the sampling interval that this does not have negative effects on accuracy.

When attempting to detect changes in the process it can sometimes be beneficial to feed the PCA algorithm with something else than plain process variables. One approach that seems to produce good results is to first normalise the data in order to reduce the effect of different wind speeds. This is achieved by using look-up tables defined earlier to replace the direct process data measurements with normalised residuals calculated with (7).

One approach is suggested in [17] where process variables are replaced with cumulative sum values. This seems to improve detection efficiency as well, but does introduce a new adjustable parameter to the system. The cumulative sum values are calculated over an observation window, so that the detection process can be run in real time.

Adding a processing window that "looks backward" a bit seems to make detec-

tion more accurate, but it adds an additional delay to the system. This delay is directly proportional to the length of the window the cumulative sum calculation uses. The results are collected in Table 3. *Correct iced* refers to percentage of iced data points detected correctly and *correct clean* refers to percentage of non-iced events detected correctly, *time to first alarm* is the time (number of measurements) from the beginning of the icing event to the first warning.

Table 3: Effect of cumulative sum buffer length on performance of the PCA-based method

buffer length	correct iced [%]	correct clean [%]	time to first alarm
10	60	80	14
20	65	92	21
30	75	96	7
40	75	97	12
50	73	95	16
60	67	84	20
70	93	61	3
80	90	72	6
90	90	76	6
100	90	73	6

From this table it is very clear that while detection accuracy goes up when the buffer length is increased, there is a point after which the the number of false positive warnings starts to go up as well. The increased number of false positive detections is caused partially by the increased reaction time to the end of icing. Longer summation time in this instance will make the signal stay above the warning level even after the measurements have dropped back to normal values. The slowness here is a flipside of increased robustness, i.e. the method will not react to quick changes in signal values. It will require the signal to stay above the warning level for a while before the final signal will react. It is also notable that almost all false positives are registered after the icing event is over. This could be interpreted as the system having some difficulties adjusting to the ending of the icing event in cases where the cumulative sum buffer is too long.

Best performing option in this case is either buffer length 30 or 70, depending on whether the emphasis is on detection speed or accuracy. Using buffer length 30 results in smaller number of correct detections, but the number of correctly detected clean incidents is very high. Higher number of correctly detected clean measurements means fewer false positive alarms. Using buffer length 70 has the highest detection rate, but it also issues a false alarm nearly 40 % of the time.

4.5 k Nearest Neighbour search

One approach for detecting changes in process behaviour is to calculate differences between variables directly. This can be done separately for each variable and then

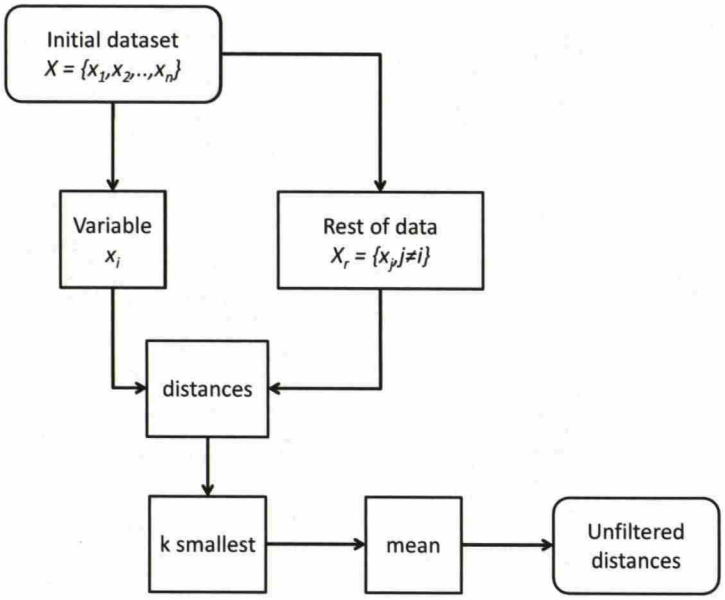

Figure 17: The construction of the reference dataset for knn-based fault detection.

by comparing and possibly combining the residuals. Another approach would be to use a multivariate method to calculate the distance of a variable from the reference set directly. Simplest approach for this would be to use k nearest neighbours search (kNN).

The basic principle is to first pick a distance measure, function that can be used to calculate the distance between two points in a set. After this the distance between the measurement point and all points in the reference dataset is calculated. From these $k(k \in \mathbb{N})$ points with the smallest distance to the new measurement are picked and and the mean of these k distances is stored. This distance is then used as a measure of distance between the measurement and the reference dataset. [36]

Proper pick of k is important when considering the effectiveness of kNN as a fault detection scheme. Larger values of k might produce more robust results, but in the context of fault detection too large values for k might obscure away the possible faults. On the other hand too small values for k might be too sensitive to outliers in data. A suitable k can be selected based on some test runs on simulated data and then the mean of k smallest distances as our final measure for the distance between measurement and the reference dataset.

The final monitored signal is the distance between the measurement and the reference data. The reference data is constructed according to flowchart in Fig. 17. First, a set of measurements is collected and then a reference dataset is constructed by calculating the distance between each point x_i and its k nearest neighbours in the data $x_j,\ j \neq i$.

The detection process is illustrated in Fig. 18. Measurements are collected from

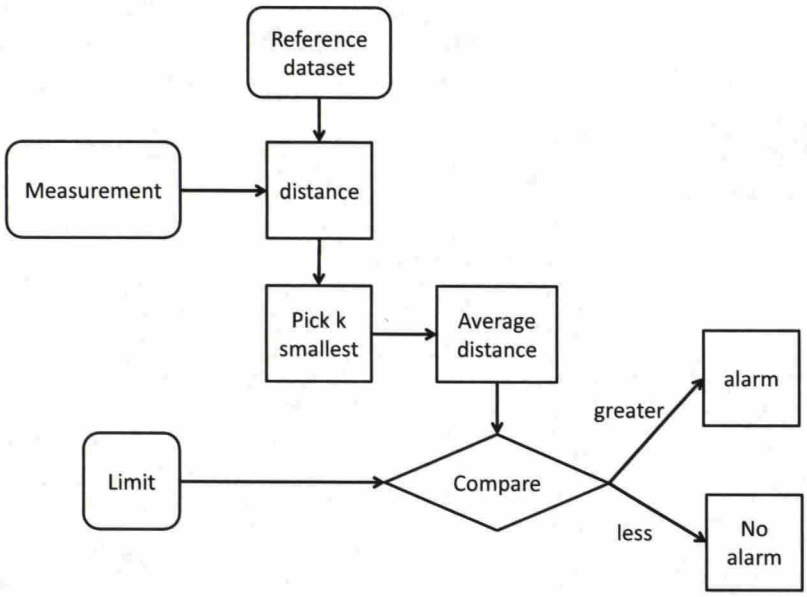

Figure 18: Fault detection procedure for knn-based fault detection.

the process and the k nearest neighbours in the reference dataset are found for each measurement and the mean distance between measurement and reference data is calculated. When a point that is too far from the reference data is detected an alarm is issued. [37]

The alarm limit for the distance can be derived from the reference dataset. One option would be to pick a percentile from the original reference data. Using a percentile limit can be interpreted as searching for data that differs from the reference data more than most points in the reference data do. This approach ignores some outliers in the reference data and produces a more sensitive detection method. Setting the limit too low will in turn increase the number of false positive signals or in the worst case can make detection impossible [37].

The basic principle for automatic limit setting is illustrated in Fig. 19. First a histogram is created for the reference dataset, that contains the distances of reference data points from the rest of the dataset. Then the right side tail of the histogram so is cut off so, that p percent of the data is below the cut off point h_r. The remainder of the data is then used as the reference dataset. Rather high values can be used for p, because the goal is to get rid of most obvious outliers. In Fig. 19 a 95 % limit was used for a simulated dataset. Prefiltering like this produces a lot more coherent dataset, as the histogram on the right hand side of the red line in Fig. 19 is a lot more symmetric.

The alarm limit is then determined in a similar manner. Either use the same upper limit h_r as for the dataset construction pick a different, higher value h_a. If $h_a > h_r$ there will be some "leg room" for measurement inaccuracies before the

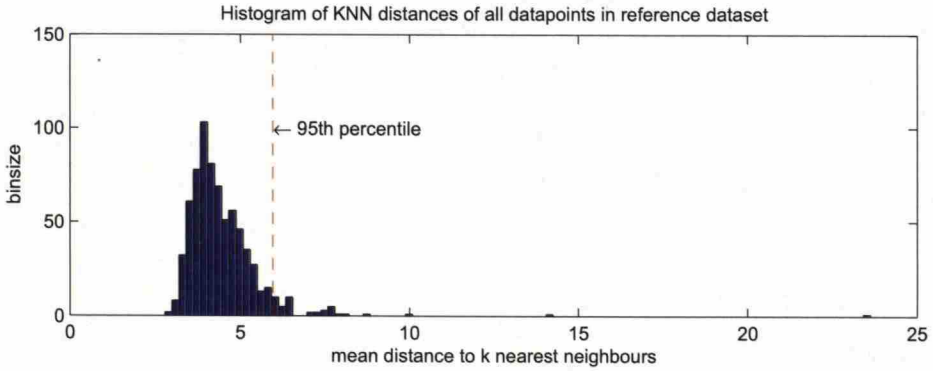

Figure 19: Creation of the reference dataset: All data to the right of the 95th percentile is ignored

method gives an alarm.

Using kNN methods produces quite noisy output signal (see Fig. 20). A more reliable result can be achieved by calculating a moving average of this output signal with (35). Smoothing lowers the output levels to some degree and makes detection slightly slower. The drop in speed is not all that significant because icing as a phenomenon is so much slower than the data sample rate. The lower signal levels might affect the detection accuracy.

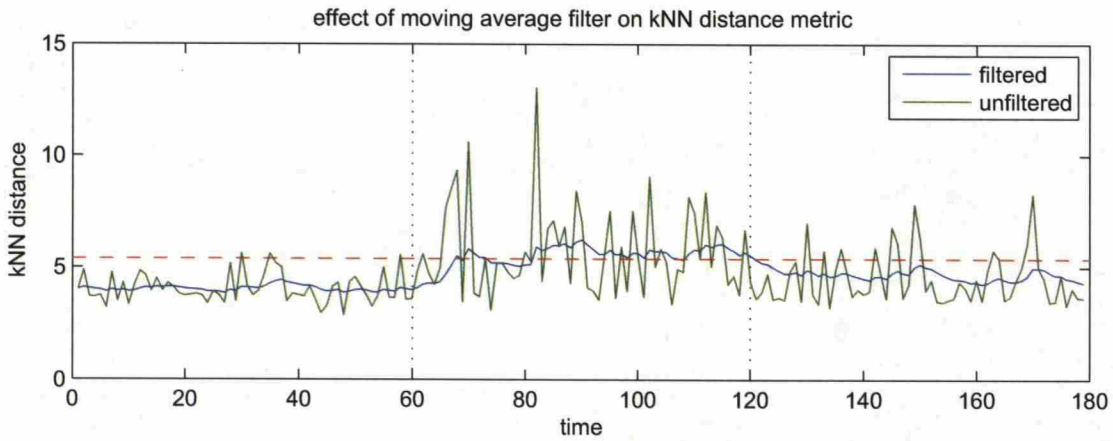

Figure 20: Effect of moving average filter on kNN detection. Dashed red line represents the alarm limit

$$d_s(t) = (1 - r)d(t) + rd(t - 1) \tag{35}$$

This leaves kNN-based methods with two degrees of freedom: value of k, and

the post filtering smoothing parameter r. Both of these need to be adjusted based on the sample rate used and the data in question. An important detail here is that kNN-based methods produce good results on non-normalised simulation data as well, where other methods do not. The only real issue here is that larger absolute values of a single measurement will end up dominating the results.

The effect of k is illustrated in Table 4. Here 4 k is the variable count, *correct iced* refers to correctly detected icing events, *correct clean* refers to correctly detected events, where there was no ice on the turbine and *time to first alarm* is the number of measurements between the start of the icing and first issued alarm. From Table 4 we can see that the detection accuracy gets better when k grows, but eventually, the performance gets worse as the number k grows.

Table 4: Effect of values of k on detection accuracy. Constant value of $r = 0.5$

k	Correct iced [%]	Correct clean [%]	time to first alarm
1	55	99	2
5	62	98	2
10	62	97	2
15	62	97	2
25	62	97	2
50	57	95	6
75	55	96	6

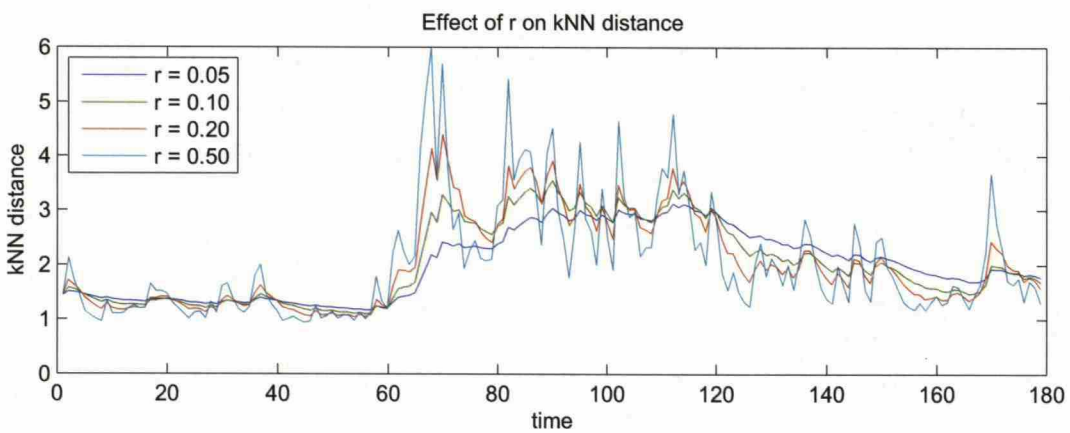

Figure 21: Effect of variable r on the filtered mean distance measure to 15 nearest neighbours in the reference data.

The effect of parameter r can be tabulated similarly. Detection statistics have been collected in Table 5 for different values of r. The table has been constructed from the same dataset as Table 4: using medium wind case and ice case 2, asymmetric ice scenario. The table does not tell the whole story, increasing the value of

Table 5: Effect of values of r on detection accuracy. When calculating these $k = 15$.

r	Correct iced [%]	Correct clean [%]	time to first alarm
0.05	63	92	22
0.10	88	97	7
0.20	85	99	6
0.40	68	96	6
0.60	60	92	2
0.90	48	91	1
0.99	52	91	1

r drops the values of the filtered distance measure, making actual detection more difficult. Figure 21 illustrates this.

The distance can be defined in many ways. Table 6 illustrates the effect of distance measurement algorithm in overall detection performance. The different distance measures in Table 6 between two vectors $x = [x_1, x_2, ..., x_n]$ and $y = [y_1, y_2, ..., y_n]$ are defined with (36) for euclidean distance, (37) for cityblock distance and (38) for Chebychev distance. Euclidean distance is defined as the absolute shortest distance between two points. Cityblock distance between two points is the sum of absolute differences in their coordinates and Chebychev distance is the maximum absolute coordinate difference between the two points. [39]

$$d_e = \sqrt{(x-y)(x-y)^T} \tag{36}$$

$$d_{cb} = \sum_{i=1}^{n} |x_i - y_i| \tag{37}$$

$$d_c = max_i(|x_i - y_i|) \tag{38}$$

Table 6: Effect of distance measure on kNN performance

distance measure	Correct iced [%]	Correct clean [%]	time to first alarm
euclidean	90	97	7
cityblock	88	97	8
chebychev	90	96	7

The differences between these are not very drastic for the simulation data used here, but the "standard" Euclidean distance measure seems to have a slight edge above the alternatives.

The number of input variables has a very visible effect on kNN accuracy as well. In Table 7 there are eight test cases with different number of process variables listed. Dropping the number of used variables causes the number of false positives to shoot up when keeping rest of the configuration constant. This shows that the number of measurements makes the algorithm more stable.

Table 7: Effect of process variable count on detection accuracy, $k = 15$ in all cases

variable count	Correct iced [%]	Correct clean [%]	time to first alarm
3	38	100	10
4	32	100	10
5	82	98	8
6	88	95	7
7	88	95	7
9	88	97	7
11	88	97	7
28	72	99	8

Using all the variables in the input data (the 28 variable case) does not produce better results. In fact, introduction of redundant data does make the detection results slightly worse. Selecting the variables beforehand is very beneficial in this case as well.

5 Simulation study results

Results drawn from simulated data are calculated using the same sets of data. Three different wind distributions for five different ice cases were used to evaluate the performance of different methods. Also the effect of available measurements on detection accuracy is illustrated here in order to evaluate the suitability of different methods for real world usage. In this section the effects of different wind and ice cases and different method specific parameters are evaluated first for each method separately. Then the performance and sensitivity of different ice detection methods are compared to each other.

Three different wind cases were needed to see if wind speed has any effect on detection accuracy. The expected result here is that detection will not be as efficient at lower wind speed because the changes in process variable values are smaller. Also, in the upper end of the power curve, the differences eventually even out, when the turbine reaches its nominal power and rotational speed. Figure 22 illustrates the drop in output power as a function of wind speed in one ice case. At lower wind speeds the change is so small that it might disappear into measurement noise.

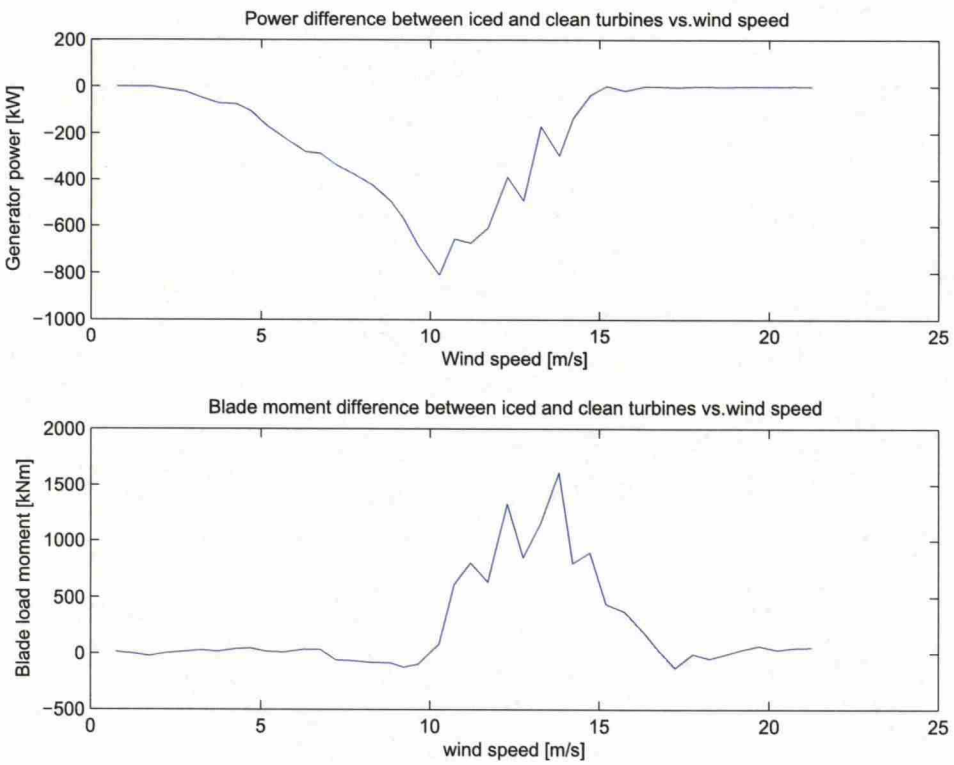

Figure 22: Effect of icing on two simulated process variables: turbine power (top) and flapwise blade moment (bottom). Ice case used here is case 3 in Table 1.

In general, the process variables used for detection here can be grouped into two groups. One group is variables that behave like power in figure 22 i.e. variables whose values drop as a result of an icing event. The other group are variables such as blade root moment in figure 22, whose values grow larger when an icing event occurs. Here difference disappears as well the at higher wind speeds.

5.1 Control charts

The goal here is to compare the detection accuracy in different wind and ice conditions and to test the response time that different kinds of control charts have to icing events of different magnitude. It is important to evaluate the effects that wind speed has on the detection accuracy of different methods. In addition to these the effect that the number of monitored variables has on multivariate methods need to be addressed.

The effects of variable numbers is interesting considering real world usage because in real world installations the number of measurements might be limited, at least when comparing to the amount of data available from the simulation model. In fact some of the simulation variables most affected by icing are things that can not be measured directly. Effect of variable number has been tested by grouping the process variables into several test cases with different variable counts. The test cases are collected in to Table 8. Measurements grouped in Table 8 are variables in simulation model output most affected by icing.

Table 8: Different variable combinations used in simulation test cases.

	Variable cases						
Measurements	3	4	5	6	7	9	11
Generator power	x	x	x	x	x	x	x
Generator speed	x	x	x	x	x	x	x
Generator torque	-	x	x	x	x	x	x
Blade pitch angle	x	x	x	x	x	x	x
Tower base moment	-	-	x	-	x	x	x
Blade root moment	-	-	-	x	x	x	x
Flapwise blade moment	-	-	-	x	x	x	x
Yaw bearing acceleration x-axis	-	-	-	-	-	x	x
Yaw bearing acceleration y-axis	-	-	-	-	-	x	x
Rotor thrust	-	-	-	-	-	-	x
Tower top displacement (fore-aft)	-	-	-	-	-	-	x

Results were calculated for all control charts introduced in Chapter 4. For the MCUSUM case it was necessary to adjust the parameter K individually for each case to get reliable detection out of the method (see chart titles in Fig. 23). This is noteworthy because smaller values of K mean that the used detection method is more sensitive to noise. This means that in less optimal conditions a larger number of variables result in better detection of anomalies as well. Regardless of attempts

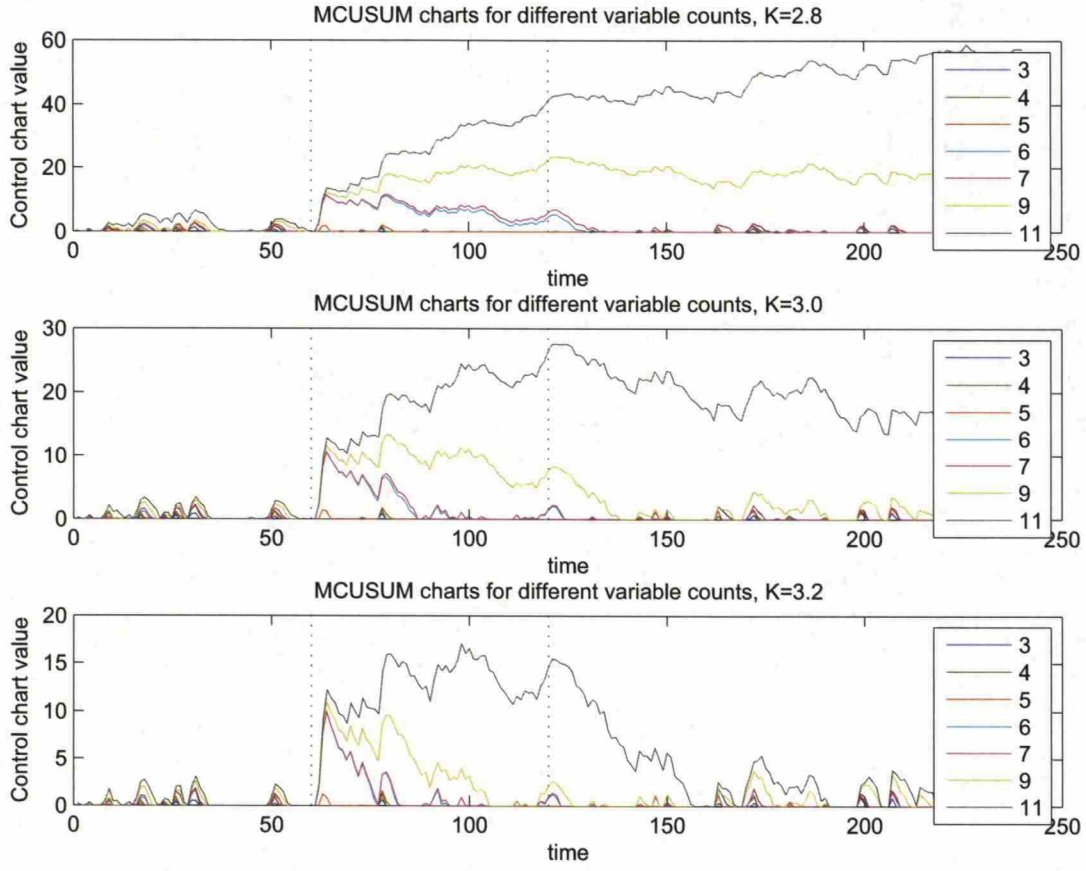

Figure 23: MCUSUM control charts for the same icing event with different variable counts and different values for parameter K.

to adjust the parameter, there doesn't seem to be a way to get reliable detection with the MCUSUM chart from just three process variables. In many cases too small value for parameter K results in charts that do not recover from an anomaly in the data. Too small values of the parameter cause some kind of saturation effect for larger variable counts that result in a control chart that does not detect the ending of the icing event.

For univariate CUSUM charts in Fig. 25 the effect of added variables is just as visible. Adding more measurements increases the values of the control chart. Interestingly, it does not affect the detection speed of the chart, but when using more variables, the difference to alarm limit value (dashed line) is bigger. Using more variables here results in a control chart that recovers slower i.e. takes a longer time to return to a safer value after the icing event is over.

Same effect is visible from the EWMA charts in Fig. 24. The signal amplitude drops significantly as the variable number drops. Here it is clearly visible as well that once the variable count drops too low, detection becomes more and more unreliable. The different signals in Fig. 24 are plotted by keeping all parameters the same across different cases, so the only differences in these plots come from changes in variable

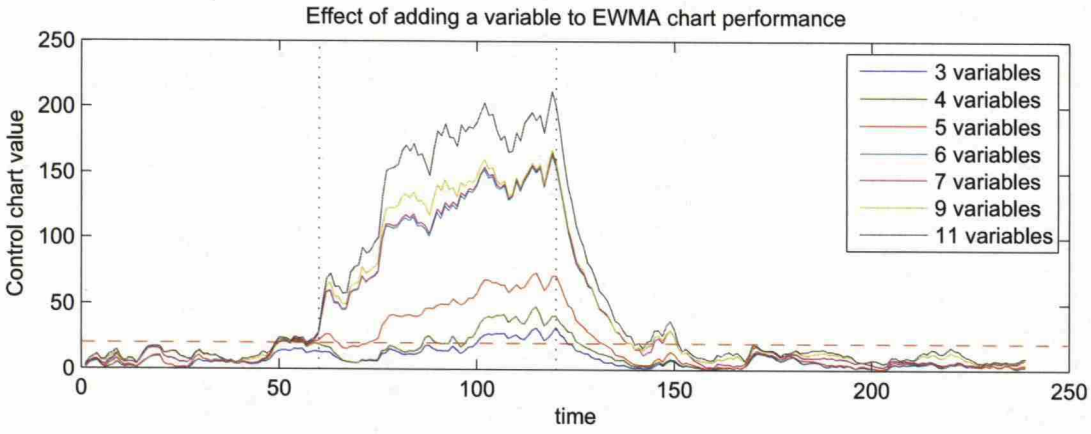

Figure 24: MEWMA control charts for the same icing event with different variable counts. Alarm limit is draw to the figure with the red dashed line. Vertical dotted lines represent the beginning and end of the icing event.

count.

In Fig. 24 the three basic variables used to plot the chart are generator power, rotational speed and the blade pitch angle. The added fourth variable was the generator torque, which is very heavily correlated with both generator power and speed. Regardless of this, adding a new measurement makes detection accuracy better even if the new measurement is somewhat redundant. The biggest difference between the different variable scenarios came from including the blade root moments. The 3, 4 and 5 variable cases are the only ones that do not include the blade root moments.

Different ice cases are clearly visible in the control charts as well as illustrated in Figs. 26 to 29. Also the difference between symmetric and asymmetric icing is somewhat visible in almost all cases, but it is so small that it doesn't really affect detection accuracy or speed. The difference grows significantly when moving to a more severe icing case, suggesting that the effect of icing will get more severe as aerodynamic penalty and ice mass increase. The less severe ice cases are more interesting because we might want to be able to detect icing as quickly as possible.

The most severe cases of icing used here seem to be separable from the data pretty much every time, using all of the different control charts. The two less severe ones are not always so clearly visible. These can be found by tuning the control charts to be more sensitive. This will cause really long recovery periods for the charts in case for more severe icing and might, in the presence of noisy data, produce a large amount of false positive detections.

This would suggest using, if at all possible, the target level of icing as a design parameter. Then to increase robustness, it could be possible to demand that the charts need to stay above the warning level for an extended period of time, instead of raising an alarm immediately after the chart crosses the warning level. Icing is relatively slow phenomenon, especially when comparing to the sample rate of the

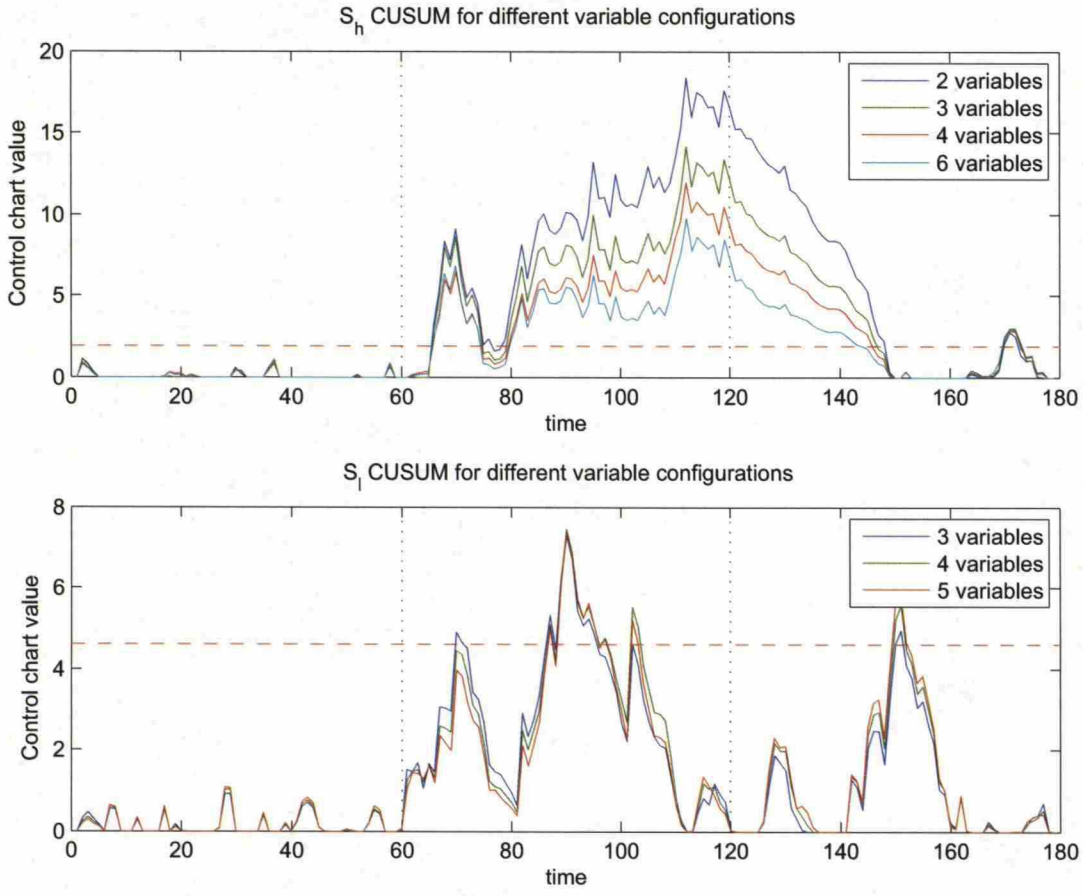

Figure 25: CUSUM control charts for the same icing event with different variable counts. Icing event between time indexes 60 and 120, alarm limits are marked on the charts with the red dashed line.

turbine control process.

The low wind speed dataset causes serious problems for some of these methods as well. The problems are most evident when using the MCUSUM chart. This is either a method-specific problem or a problem with the dataset, most likely a combination of both. The fact that other methods seem to be able to handle this dataset without issues, would suggest that it is most likely an issue with the method in question. The problematic part of the dataset has very low wind speed, so the problems are most likely caused by the relatively small values of the process variables.

The low wind speed dataset is somewhat more difficult to handle because the changes in process variable values are so small that legitimate differences might get lost in process noise even in the simulated case where the data is very coherent. In a real world application the data will have significantly more noise in it, so the methods tested here should behave very well with simulated data to have hope of performing with real world measurements.

The detection statistics for each control chart for all wind and ice cases are listed in Tables 9 to 17. In these tables the listed values are the percentage of correctly

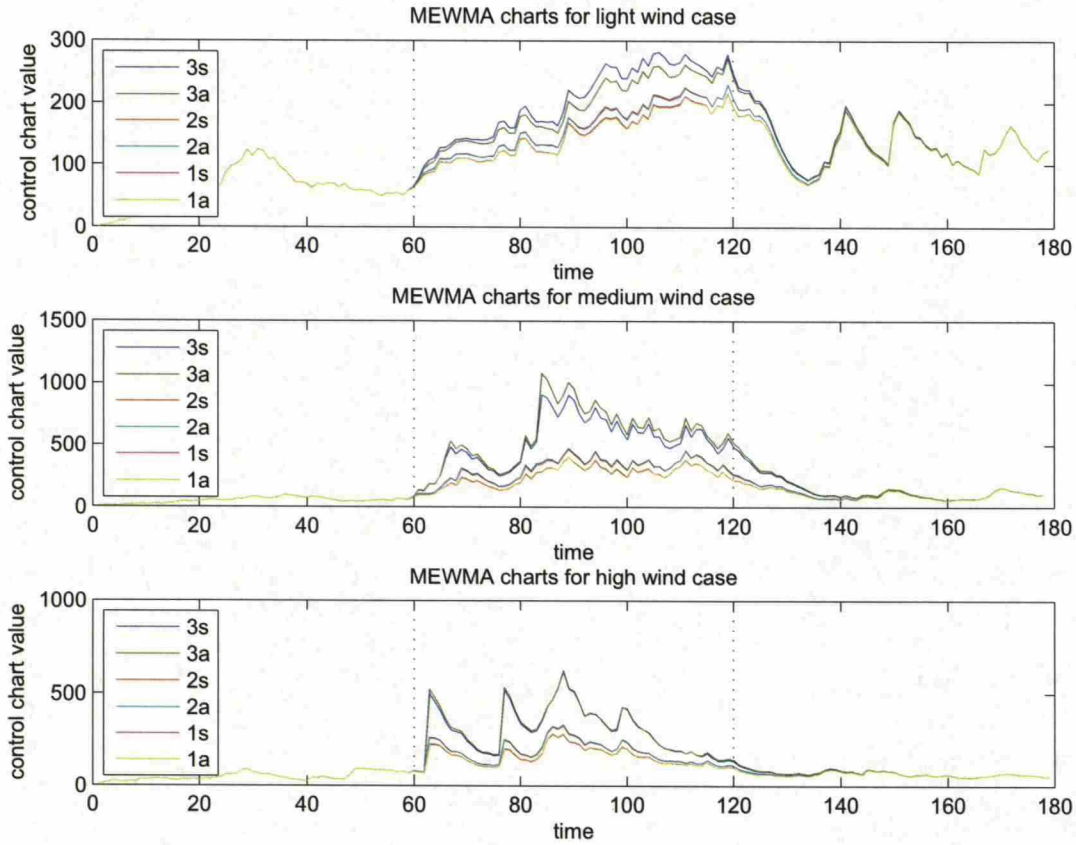

Figure 26: MEWMA control charts with different ice cases. Ice cases defined in table 1. a represents asymmetric and s symmetric icing

Table 9: The performance of MEWMA control charts in the low wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	83	54	10
3a	80	55	12
2s	67	61	18
2a	65	61	21
1s	57	63	21
1a	57	63	21

detected iced datapoints, the percentage of clean (not iced) datapoints detected correctly and the detection speed repesented as the time (number of sample) between the start of icing and the first issued alarm. In an ideal situation, the values of the first two columns need to be as high as possible and the value of time to first alarm needs to be as low as possible. Ice cases are defined in Table 1.

MCUSUM charts require that the parameter K is adjusted manually for each wind case, and regardless of the parameter adjustment, the low wind test case was

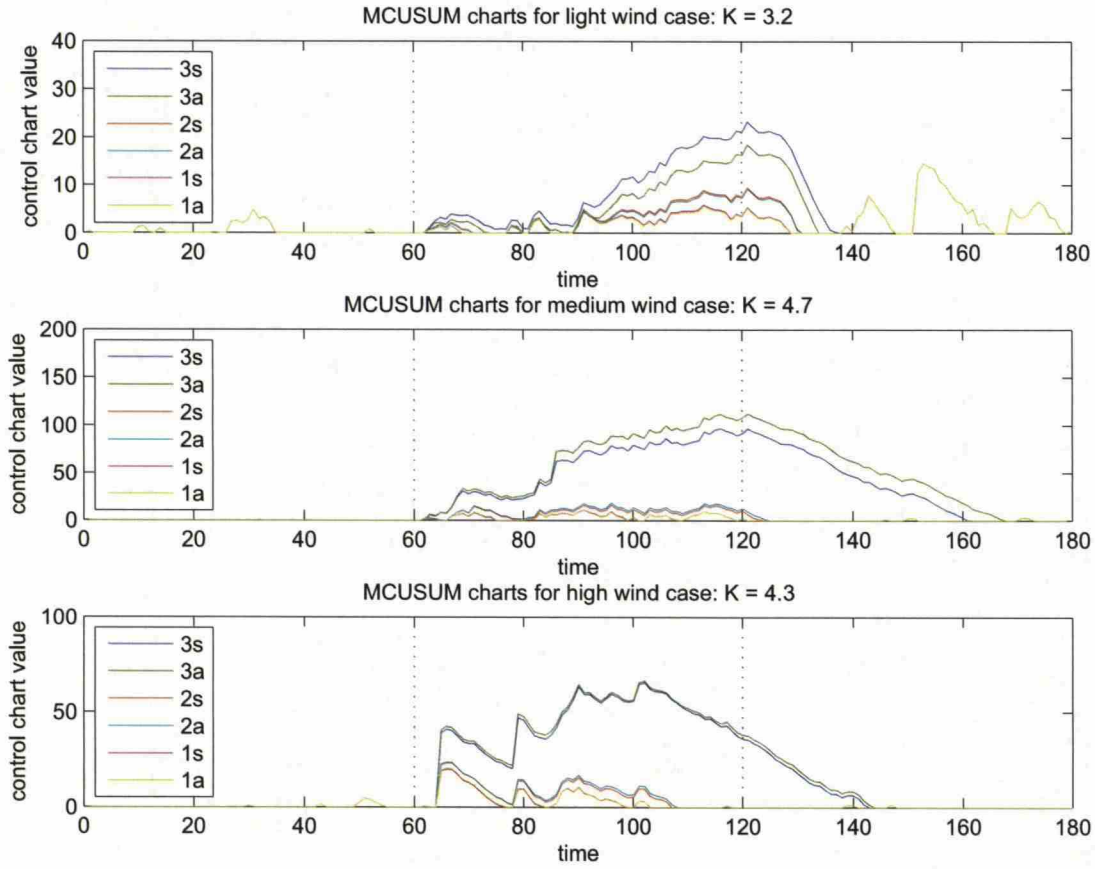

Figure 27: Multivariate CUSUM control charts for different ice cases. Ice cases defined in table 1. a represents asymmetric and s symmetric icing

Table 10: The performance of CUSUM control charts in the low wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	42	62	11
3a	25	62	11
2s	18	64	30
2a	17	64	30
1s	18	64	30
1a	15	64	30

too difficult to detect properly. For the univariate CUSUM chart the detection results are good when using the same configuration for the low and high wind test cases, but for the medium case the detection accuracy can be improved by increasing the value of the parameter k in (11) or (12). Also for the MCUSUM chart the parameter adjustment effected accuracy between different ice cases, it seems to be difficult to find a setting that works well for both the most severe ice case and

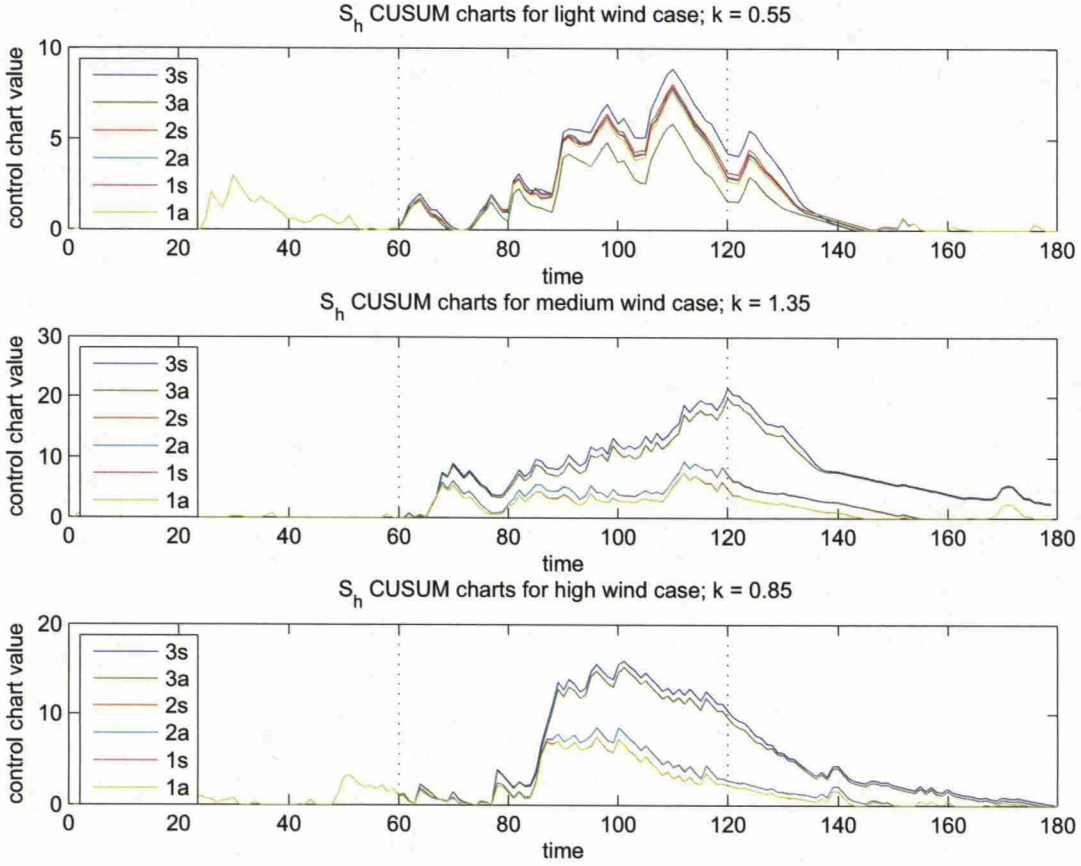

Figure 28: S_h CUSUM control charts for different ice cases. Ice cases defined in table 1. a represents asymmetric and s symmetric icing

Table 11: The performance of MCUSUM control charts in the low wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	25	90	45
3a	13	92	52
2s	0	99	–
2a	0	99	–
1s	0	99	–
1a	0	99	–

the least severe ice case at the same time. This is more problematic than wind dependency because naturally there is no real way to determine the severity of icing beforehand. Detection rates with these control charts are decent enough, but the wind speed dependence of the parameters makes CUSUM charts lot less practical than MEWMA chart. Also MEWMA charts do allow an easy method for calculating an upper limit for detection as well, which makes it a lot better option for practical

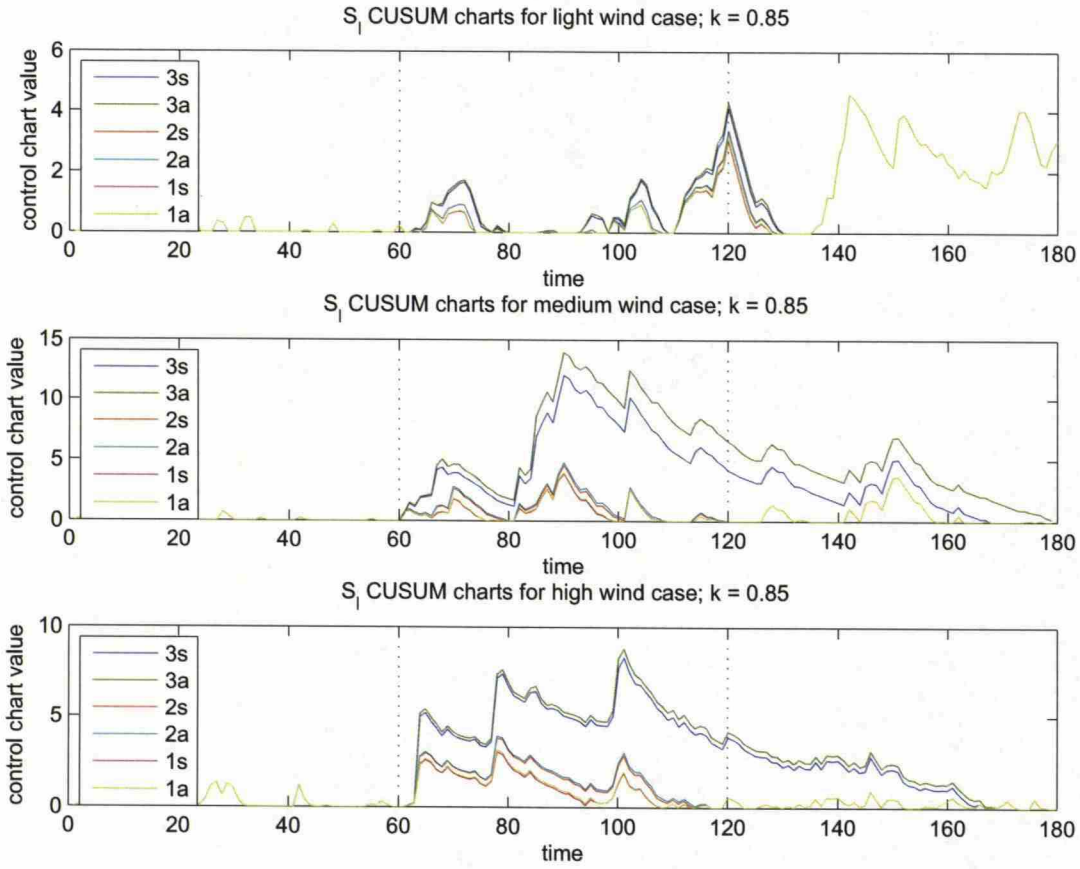

Figure 29: S_l CUSUM control charts for different ice cases. Ice cases defined in table 1. a represents asymmetric and s symmetric icing

Table 12: The performance of MEWMA control charts in the medium wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	93	62	4
3a	93	62	4
2s	88	75	7
2a	88	73	7
1s	88	77	7
1a	88	77	7

use.

The wind speed dependency of the CUSUM chart parameter seems to be related to the observation that the differences from norm are bigger at medium wind speeds. Because the possible changes are bigger at those wind speeds the detection method needs to be adjusted accordingly. Still, the parameter need to be set to a value that

Table 13: The performance of CUSUM control charts in the medium wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	83	85	6
3a	82	86	6
2s	42	98	6
2a	42	98	6
1s	27	98	6
1a	27	98	6

Table 14: The performance of MCUSUM control charts in the medium wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	92	58	5
3a	93	50	3
2s	75	95	8
2a	80	94	8
1s	40	100	9
1a	43	100	9

is comparable to the size of the expected change in the variable value. This might be hard to determine beforehand, without reference data for different icing scenarios or a realistic simulation model.

Table 15: The performance of MEWMA control charts in the high wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	93	95	4
3a	93	95	4
2s	85	99	4
2a	85	99	4
1s	80	100	4
1a	80	100	4

Table 16: The performance of CUSUM control charts in the high wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	93	56	4
3a	93	55	4
2s	90	81	4
2a	90	80	4
1s	82	86	4
1a	82	86	4

Table 17: The performance of MCUSUM control charts in the high wind test case

ice case	correct iced [%]	correct clean [%]	time to first alarm
3s	92	86	5
3a	92	83	5
2s	43	100	5
2a	47	100	5
1s	25	100	5
1a	25	100	5

5.2 PCA

When using PCA-based methods for detection there are a few factors that contribute to detection accuracy. The most important one is the number of principal components used for detection. Other factors that contribute to detection statistics are the method used to combine the Q and T^2 statistics and the length of the possible pre filtering buffer (see Table 3).

The two different statistics both produce a boolean alarm signal. These need to be combined to get the final output of the ice detection method. The two signals can be combined by issuing an alarm when both of them are above the alarm limit (equivalent to combining the two signals with a logical AND-operation) or issuing an alarm when either one crosses its warning level (logical OR).

The differences of these two operations are collected in Tables 18 and 19. Results in both of these tables illustrate the effect that increasing the number of principal components has on detection performance. In these tables *correct iced* refers to share of correctly detected iced datapoints *correct clean* means percentage of correctly detected non-iced samples and *time to first alarm* refers to the number of samples between the beginning of icing and the first issued alarm.

One thing that can be seen from Tables 18 and 19 is that the operation used for chart combination affects the number of required principal components needed to reach a certain performance level. On the other hand increasing the number of principal components and using an OR-operation causes the number of false positive detections to spiral out of hand. Seemingly, at least with the simulated data, demanding that both charts need to cross the line before issuing an alarm

Table 18: Effect the number of principal components has on detection accuracy. Statistics combined with OR-operation.

λ	Correct iced [%]	Correct clean [%]	detection speed
3	77	91	11
4	83	84	10
5	87	75	8
6	88	57	7
7	88	50	7
8	95	41	1
9	100	33	1
10	100	27	1
11	100	20	1

Table 19: Effect the number of principal components has on detection accuracy. Statistics combined with AND-operation.

λ	Correct iced [%]	Correct clean [%]	detection speed
3	0	99	–
4	62	97	23
5	65	98	11
6	78	87	11
7	83	87	10
8	83	85	10
9	87	86	8
10	85	86	9
11	85	87	9

seems to produce a significantly more robust detection method, even though it does not seem to be able to find all iced data points and requires slightly larger number of principal components to produce similar performance.

Another clear conclusion that can be drawn from Tables 18 and 19 is that increasing the number of principal components makes the detection accuracy better only to a certain degree. There is a point after which increasing the dimension of the principal component basis does not make the method perform any better.

The effect of ice mass is illustrated in Figs. 30, 31 and 32 for all three wind distributions defined earlier. These graphs illustrate the values of Q and T^2 statistics for different ice cases when all method parameters and wind time series are kept constant. In all cases the icing occurs between time points 60 and 120. Icing alarm is issued at any time both Q and T^2 charts are above the alarm limit illustrated with red dashed line. Ice cases have been defined in Table 1, with the addition that a refers to asymmetric and s refers to symmetric icing.

The detection statistics have been collected in Tables 20, 21 and 22. In these tables *correct iced* refers to share of correctly detected iced datapoints *correct clean*

Table 20: Different ice cases for the low wind test case.

Ice case	Correct iced [%]	Correct clean [%]	detection speed
3s	73	87	16
3a	67	88	20
2s	23	91	37
2a	18	90	49
1s	18	91	49
1a	18	90	49

Table 21: Different ice cases for the medium wind test case.

Ice case	Correct iced [%]	Correct clean [%]	detection speed
3s	87	85	9
3a	88	85	8
2s	78	87	11
2a	80	87	11
1s	68	87	14
1a	68	87	14

means percentage of correctly detected non-iced samples and *time to first alarm* refers to the number of samples between the beginning of icing and the first issued alarm. The results clearly show that PCA is significantly slower than e.g. control chart based methods in issuing the alarms. Also the accuracy of PCA-based methods seems to be at above 50 % only in the medium wind test case and the most severe ice case in the light and high wind test cases. In other words in 8 of the 18 test cases the PCA-based method missed more than half of the iced samples. This was either due to slowness of the method or due to general method insensitivity.

The ice cases affect the detection accuracy as well. The PCA methods seem to work best for the medium wind test cases. On higher wind speeds, the detection accuracy is lower than in the low wind case. This is especially true for the milder ice cases. For example, in the high wind test case, the detection accuracy drops to approximately 15 % from 50 % for the more severe ice case. This is in line with

Table 22: Different ice cases for the high wind test case.

Ice case	Correct iced [%]	Correct clean [%]	detection speed
3s	52	100	26
3a	52	100	26
2s	15	100	36
2a	15	100	36
1s	13	100	36
1a	13	100	36

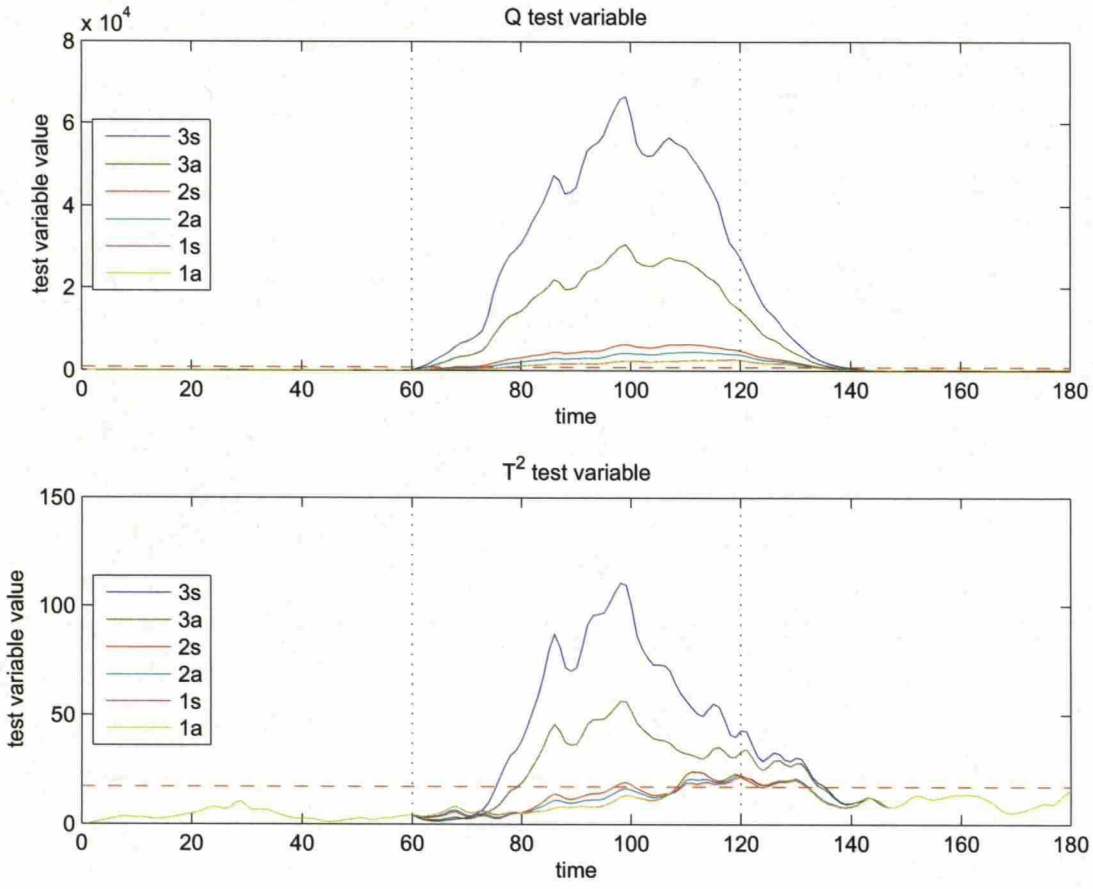

Figure 30: Test variables for different ice cases for the low wind case.

the previous observation that the differences in process variable values tend to trend towards zero as wind speed grows (see Fig. 22).

As a result of this the detection here works best near the maximum (blade moment) or minimum (generator power) of the curve illustrated in Fig. 22 and becomes more and more uncertain at lower and higher wind speeds. In the middle of the range PCA-based detection works very well for the simulation data, even for the lighter ice cases the detection is reliable and fast.

When considering real world usage there are several possible sources for issues when using these methods. Tuning a PCA based system was a bit complicated and requires several independently tuned steps to make it work with simulated data. This can cause problems when operating on actual measurements, because there might not be any data from an actual icing incident to use as a reference in order to get the algorithms tuned in a way that detection is overall possible. Also the added complexity and computational overhead of PCA will not necessarily improve detection accuracy when comparing to simpler control charting methods, especially when the number of used process variables is relatively small as compared to 27 in the full simulation data.

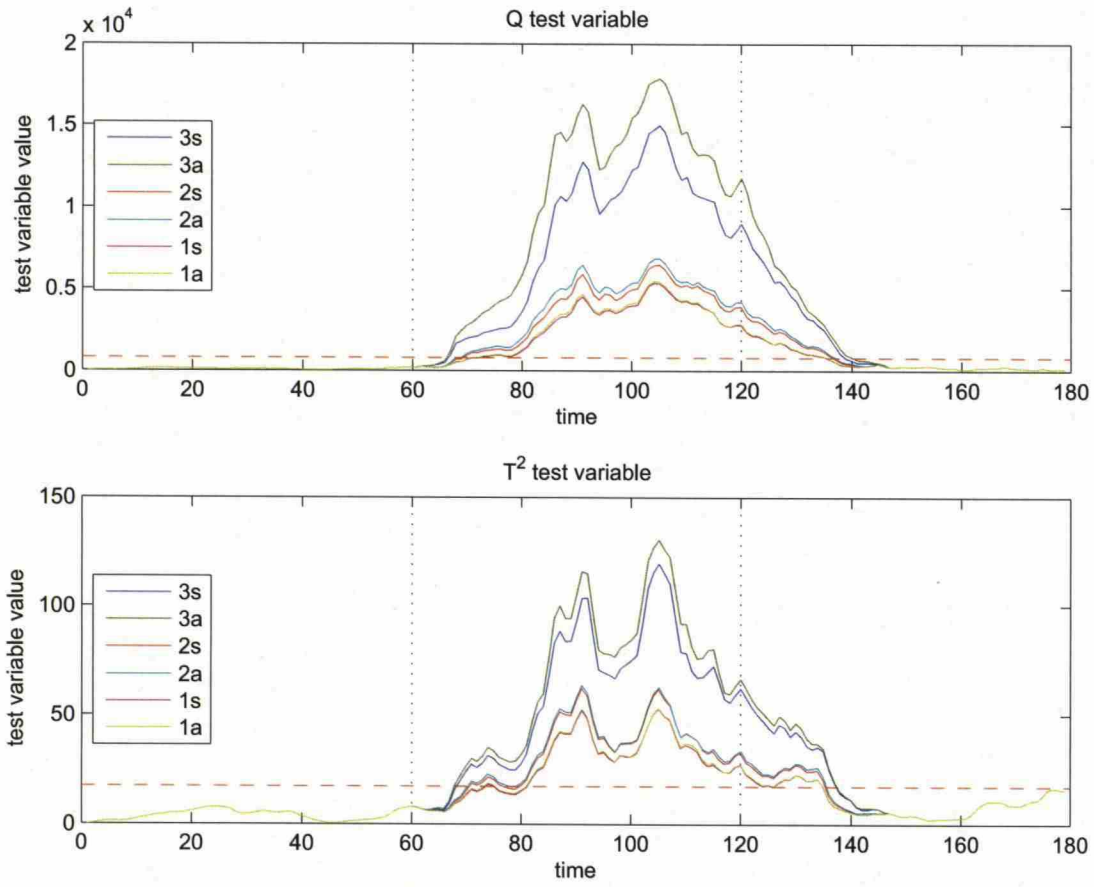

Figure 31: Test variables for different ice cases for the medium wind case.

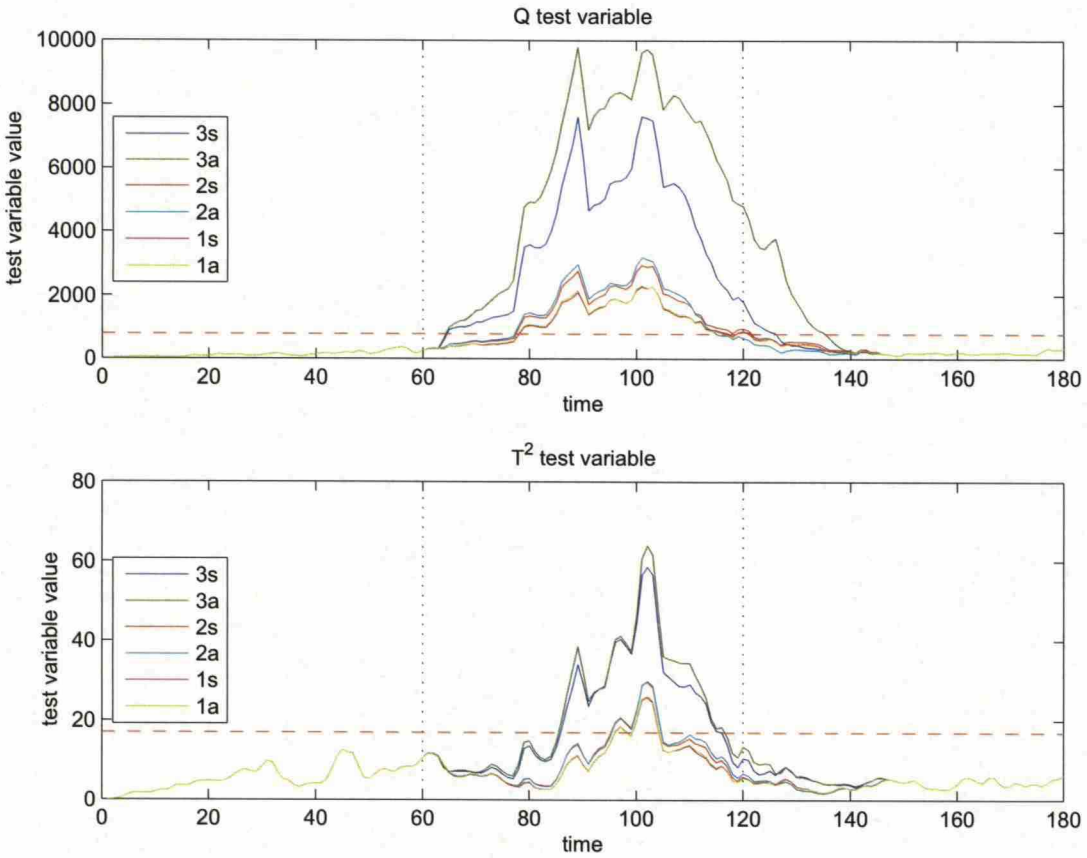

Figure 32: Test variables for different ice cases for the high wind case.

5.3 kNN

Using kNN methods to calculate the distance of measurements from the reference data results in similar behaviour as described earlier for other methods. Smaller ice masses and smaller wind speeds make detection harder. The medium wind test case produced best results here as well, just like with all other methods tested.

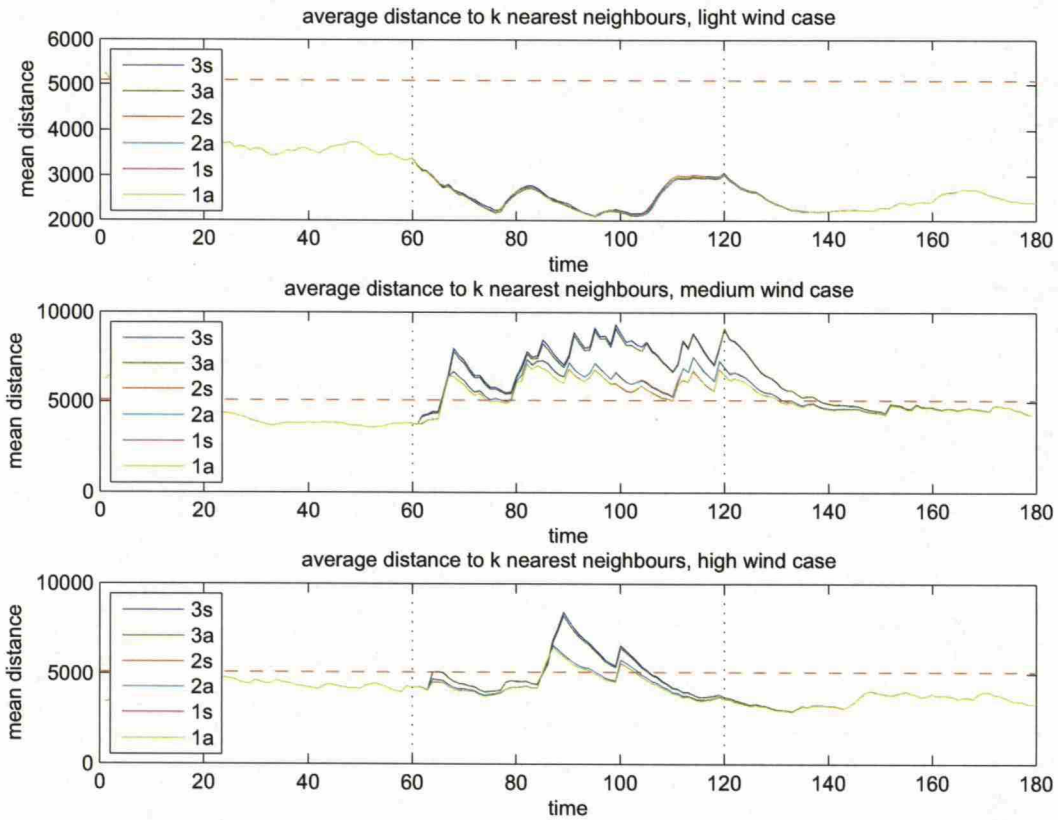

Figure 33: Effects of changes in ice mass on kNN distance for different wind cases. k is kept constant at 50.

Figure 33 illustrates the average distance from test case measurements to reference data set. Interestingly this method does not produce any warnings for the low wind test case. For the high wind case the same configuration produces only a short peak above the warning line. There is also a rather lengthy delay between the icing event and first detection. On the other hand, detection works just fine for the medium wind test case.

Notable point here is that these results have been calculated, unlike all others, for non-normalised data. This method finds icing events from normalised data as well, but the output looks slightly different. The absolute values of the distance measure are significantly smaller. Not using normalised data is problematic because of differences in absolute values of variables. The worst case scenario is that a single measurement can dominate the final results and all the others might become irrel-

evant. Normalising the data removes the possibility of a single variable dominating the results. This is what happens with the simulation data in the test case as well. There is a single variable with significantly larger absolute values than the rest and as a result, the resulting graphs pretty much represent the behaviour of that one measurement.

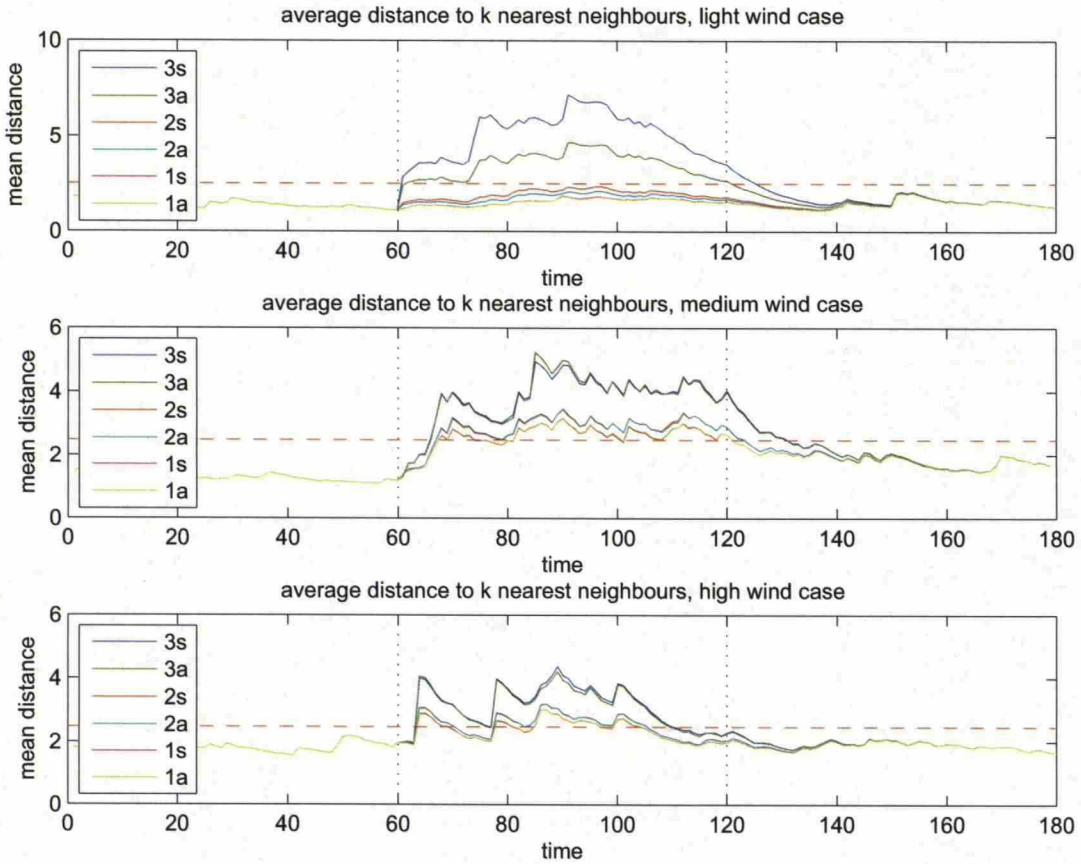

Figure 34: Effects of changes in ice mass on kNN distance for different wind cases for normalised data. k is kept constant at 15.

The results for different wind and ice cases are collected in Tables 23, 24 and 25. In these tables *correct iced* refers to share of correctly detected iced datapoints *correct clean* means percentage of correctly detected non-iced samples and *time to first alarm* refers to the number of samples between the beginning of icing and the first issued alarm. These tables show that the kNN method performs at an acceptable level except in the light wind test case. In the other two the detection is fast and relatively accurate. The small number of false positive alarms (large percentage of correctly detected clean samples) is also encouraging.

When using normalised data the results are slightly different, the absolute values of the distance measure are smaller, but overall the detection accuracy is better. The results are illustrated in Fig. 34 Also when using normalised data, the method is

able to detect icing in the low wind test case as well, but only for the most severe case of icing. Also, the detection accuracy is better for the high wind test cases. This is clear evidence that data normalisation does benefit the method rather significantly and is useful addition to the detection method, regardless of the added complexity.

Table 23: Effectiveness of KNN-based detection for different ice cases in the light wind test case.

Ice case	Correct iced [%]	Correct clean [%]	time to first alarm
3s	98	96	1
3a	97	99	2
2s	0	100	–
2a	0	100	–
1s	0	100	–
1a	0	100	–

Table 24: Effectiveness of KNN-based detection for different ice cases in the medium wind test case.

Ice case	Correct iced [%]	Correct clean [%]	time to first alarm
3s	90	90	6
3a	88	90	7
2s	87	97	8
2a	87	97	8
1s	72	98	8
1a	75	98	8

Table 25: Effectiveness of KNN-based detection for different ice cases in the high wind test case.

Ice case	Correct iced [%]	Correct clean [%]	time to first alarm
3s	77	100	4
3a	75	100	4
2s	57	100	4
2a	58	100	4
1s	43	100	4
1a	45	100	4

A clear difference between kNN and other methods discussed here is the computing time required to create the reference dataset. The method described here requires calculating the distances between every point in the reference dataset and every other point in the dataset. This means calculating the kNN distance $N + 1$ times when creating the reference dataset. Here N is the size of the reference

dataset. Other method would require looping through the dataset only once for the lookup table. This makes initialising the kNN method significantly slower, but the initialisation is a step that needs to be done only once, so the difference is not that relevant. [38]

The results for this particular dataset are somewhat promising, but performance of kNN seems to be affected by wind speed. This is the same wind speed related behaviour that was detected earlier for other methods as well. Detection accuracy in all these methods is related to the size of the anomaly and the change in variable values is smaller at smaller wind speeds. There exists a "sweet spot" for all these methods: a wind speed range somewhere close to the rated wind speed where these kinds of methods seem to work best. Above and below this range the detection becomes harder and less reliable.

5.4 Conclusions from the simulation study

When comparing the results from different methods the results are somewhat mixed. Overall, each of the examined methods can detect icing in the simulated case. The real differences come from the overall robustness and stability of the examined methods: How well these methods work on different ice cases and even different wind distributions?

The behaviour is similar in all cases, methods work better at wind speeds somewhat below the rated speed. More severe icing produces a larger response in the control charts or other measures. Normalising the data produces better outputs in all analysed cases as well. Normalisation makes it easier to compare the relative impact of different measurements on the final control chart and it removes the requirement of analysing the units used to save a variable.

The detection accuracy of different methods is plotted in Figs. 35, 36 and 37. All figures represent the detection accuracy of different methods in different ice cases for a certain wind distribution. Generally it seems that the MEWMA control charts work well in all cases, even though it is more or less the simplest method to implement. It seems to be the only method that works well in all test cases.

It seems that detection accuracy in these simulations is a very visibly related to wind speed. The medium wind test case produces best results overall, and even in the higher wind tests the performance of the CUSUM control charts is rather good. The kNN and PCA methods and the MCUSUM control chart have serious problems with ice cases 1 and 2. Only in the best-case scenario (the medium wind test case) are these two able to capture more than half of the icing incident in these two ice cases. The performance in ice case 3 is at a comparable level in all wind scenarios.

There does not seem to be a significant difference between symmetric and asymmetric icing. Asymmetric icing is slightly easier to detect and there is a slight difference in e.g. control chart values between symmetric and asymmetric case (asymmetric is without exceptions worse). But the differences are rather small. The methods used here monitor changes in the variable values and based on this the difference in ice distribution does not affect the result too much.

The detection speed is calculated as samples between the start of icing and the

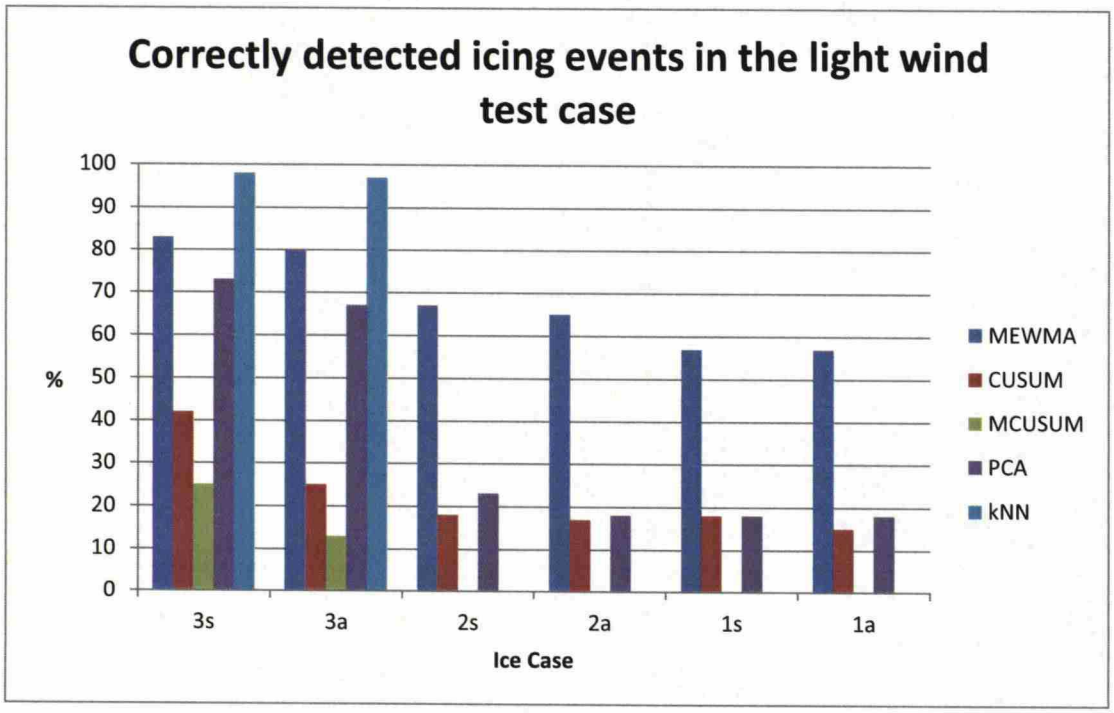

Figure 35: Detection accuracy of different methods in different ice cases. Ice cases defined in table 1, s refers to symmetric icing a to asymmetric.

first issued alarm. These are shown in Figs. 38, 39 and 40. Generally all methods perform in a relatively similar way, but PCA seems to always be the slowest of the bunch in all cases and the MCUSUM control chart and kNN methods do not seem to work very well in the light wind test case.

Both multivariate control charts require that the variables are picked manually beforehand in a way that the variable matrix only contains variables that are actually affected by icing. This needs to be done in order to avoid ending up with a covariance matrix that is too close to singular. A singular covariance matrix would make calculating the multivariate control charts impossible because the inverse of the covariance matrix does not exist in (14) and (16). This is caused by existence of measurements not affected by icing in the data.

The univariate CUSUM charts go a step further. Both of the charts detect only differences in one direction: S_h reacts to increases in variable mean and S_l reacts to events where variable mean is less than the reference value. Successful detection requires if values used for these charts are picked according to expected behaviour.

The false positive results observed in the simulation case were mostly a result of slower than required recovery. Meaning that the ice alarm stays on after the icing incident had ended, this could also be a feature of the dataset, because the changes in the ice mass were sudden steps. On the other hand missed detections were almost entirely, with few exceptions due to a delay between the start of icing and the first alarm. After the alarm was activated, it mostly stayed on until the icing event had

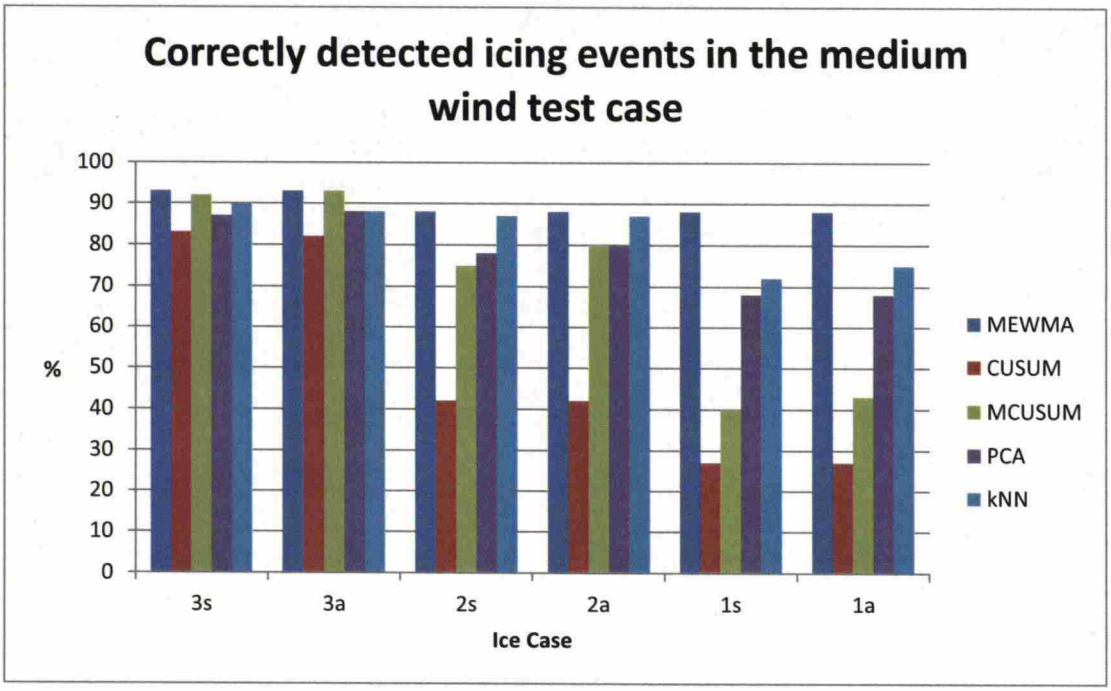

Figure 36: Detection accuracy of different methods in different ice cases. Ice cases defined in table 1, s refers to symmetric icing a to asymmetric.

ended.

Other methods discussed here do not require manual variable selection, making these methods easier to implement. Performance of kNN methods does improve, when redundant variables i.e. variables not affected by icing are removed. Not requiring variable selection means that it is possible to implement the detection method by simply using all available measurements as inputs. PCA will do the variable selection mathematically, but the number of principal components needs to be adjusted when variables are added.

The complexity of the chosen method does affect the usability of the methods. The control charts are rather straightforward to implement and fast to use in different environments; using PCA requires some advance preparation and the added step of the principal component generation. Using PCA-based detection in practice also means that every time the reference dataset is updated, the principal component scores need to be updated as well. This adds an additional level of complexity and an extra step in the implementation. The biggest issue in using kNN based detection as described here is the slowness of the reference dataset creation. Every time a new set of measurements is added to the reference dataset, the method needs to calculate the distance to every other point in the dataset. As a result the addition of new data will become significantly slower as the reference dataset grows.

The tests with simulated dataset show that detection is possible using any of these methods. There are differences in accuracy, speed and overall usefulness, but

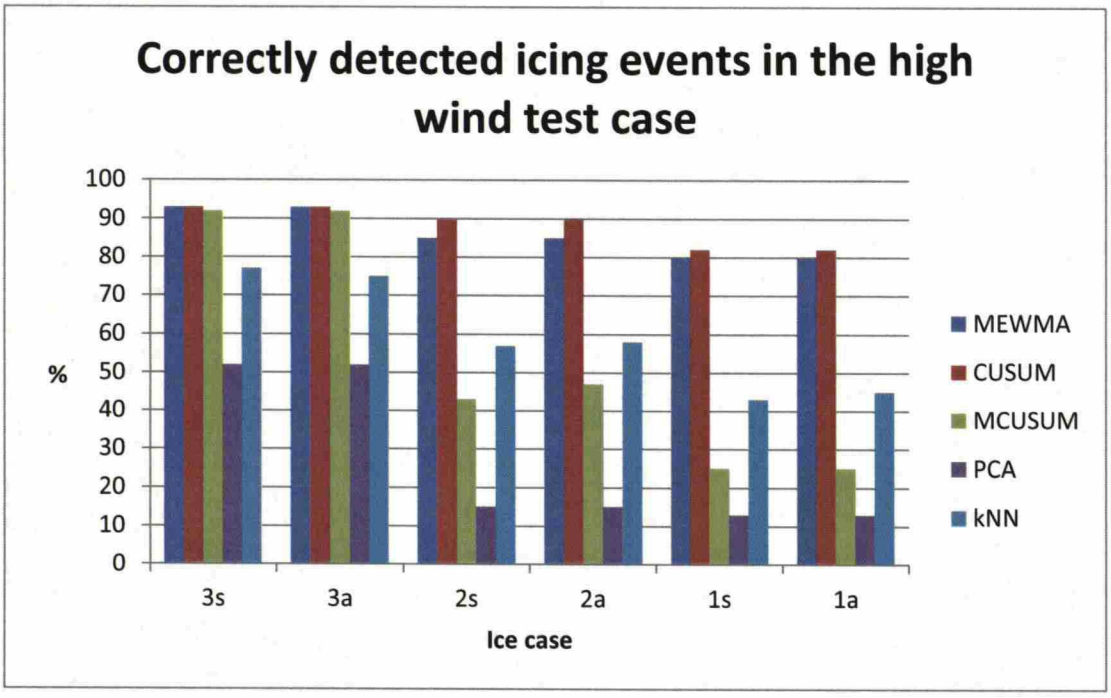

Figure 37: Detection accuracy of different methods in different ice cases. Ice cases defined in table 1, s refers to symmetric icing a to asymmetric.

they all seem to work. On the other hand it is hard to say how well these methods will work in real life scenarios, because the simulation data has very little noise and is overall very well organised.

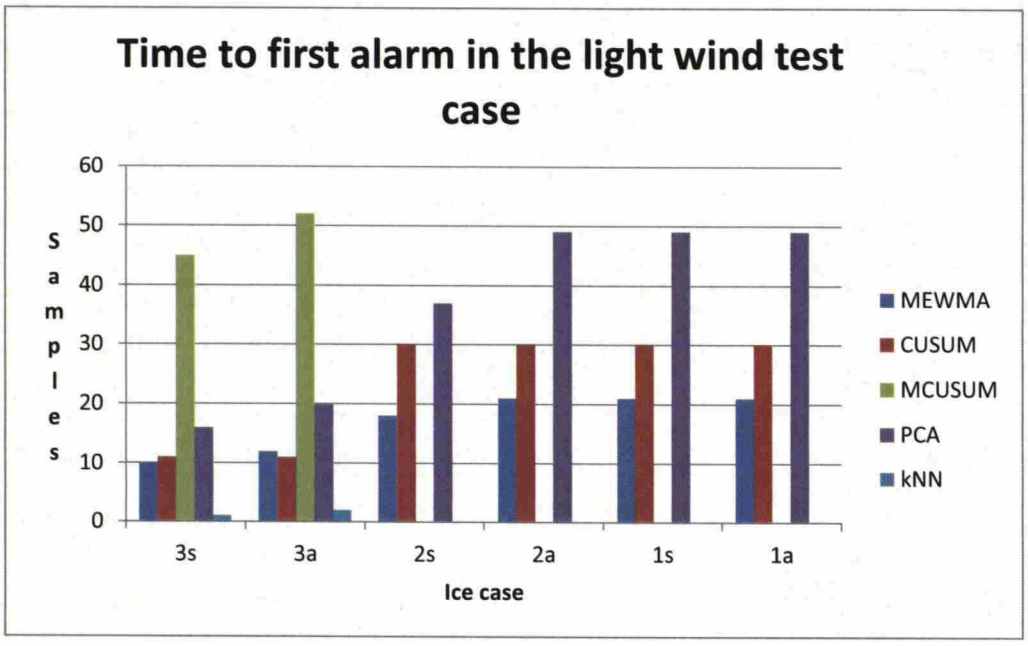

Figure 38: Detection speed of different methods in different ice cases. Ice cases defined in table 1, s refers to symmetric icing a to asymmetric.

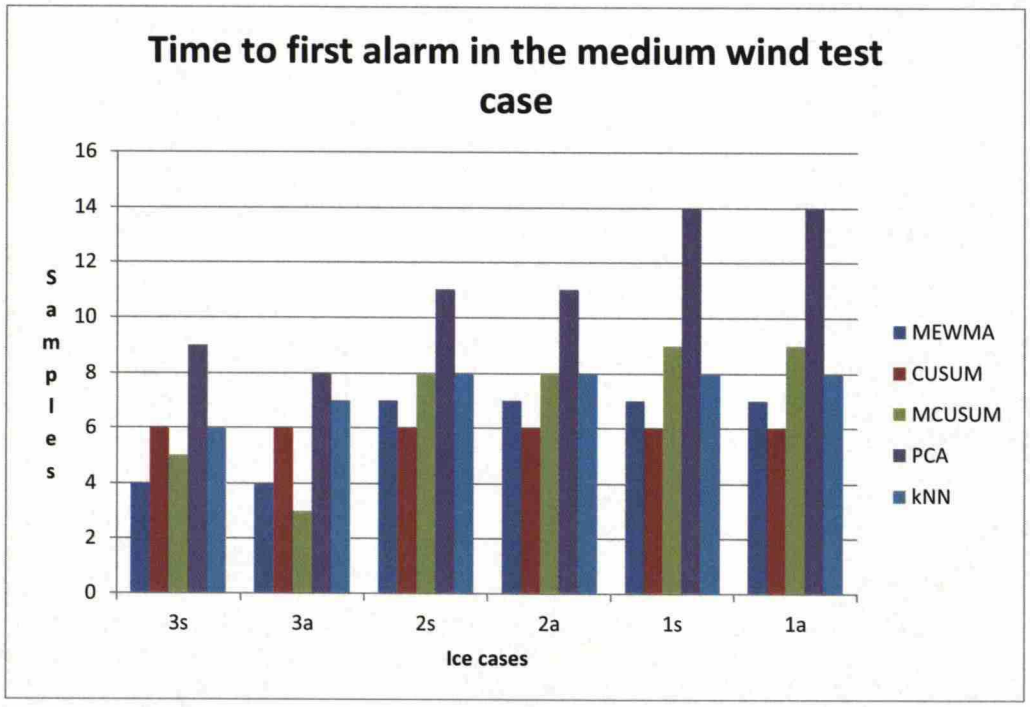

Figure 39: Detection speed of different methods in different ice cases. Ice cases defined in table 1, s refers to symmetric icing a to asymmetric.

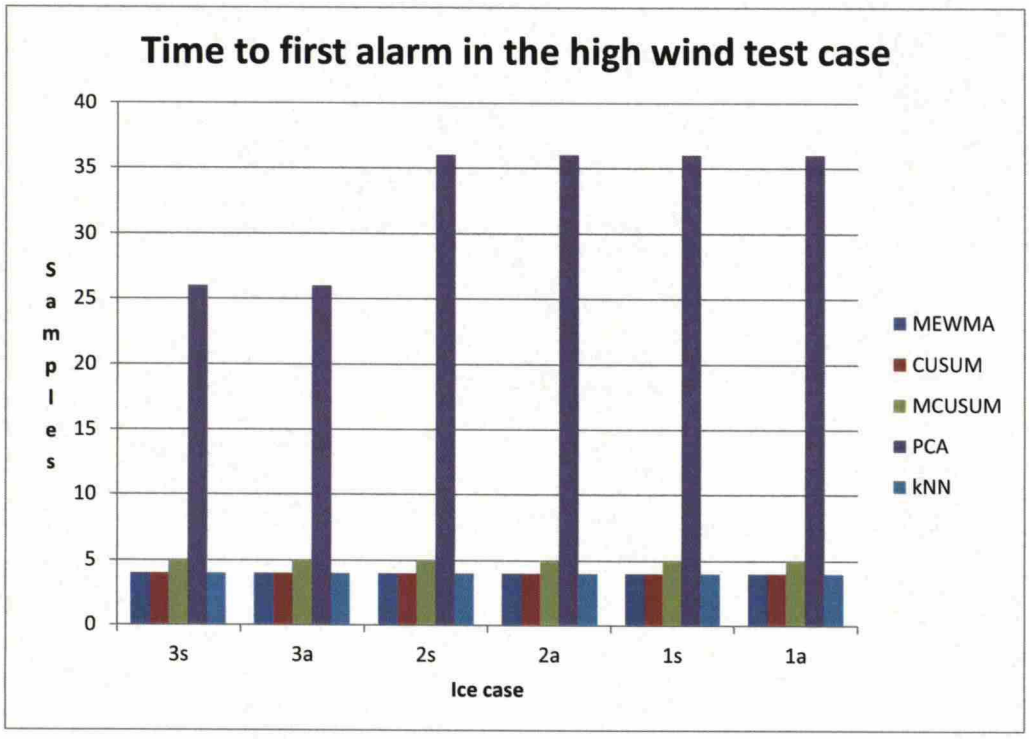

Figure 40: Detection speed of different methods in different ice cases. Ice cases defined in table 1, s refers to symmetric icing a to asymmetric.

6 Case Study

In order to test the performance of these methods in real world usage, a case study with real turbine data was conducted. The test case data was collected from a multi MW turbine in northern Finland over 2010-2012. There was an ice sensor mounted at the top of the nacelle, but the ice sensor data is only available for the final year of the data.

There are fewer available measurements in the real data than there were in the simulations. The available measurements in this dataset are generator power and rotational speed, blade angles, outside temperature and wind speed and direction. All measurements in this set are ten minute averages. This set of measurements is enough to test whether the methods described earlier work in real world scenario or not.

The data was divided into two sets according to temperature: a warmer reference set and a colder set that was used as a detection target. The most obvious outliers were removed from the reference dataset beforehand to ensure a more reliable performance. The filtering is demonstrated in Fig. 41, most obvious outliers were removed to ensure slightly better detection accuracy. This kind of outlier removal was only done to the reference dataset. The reason for limiting the pre-processing data clean-up to the reference dataset was to avoid the possibility of filtering out possible icing incidents.

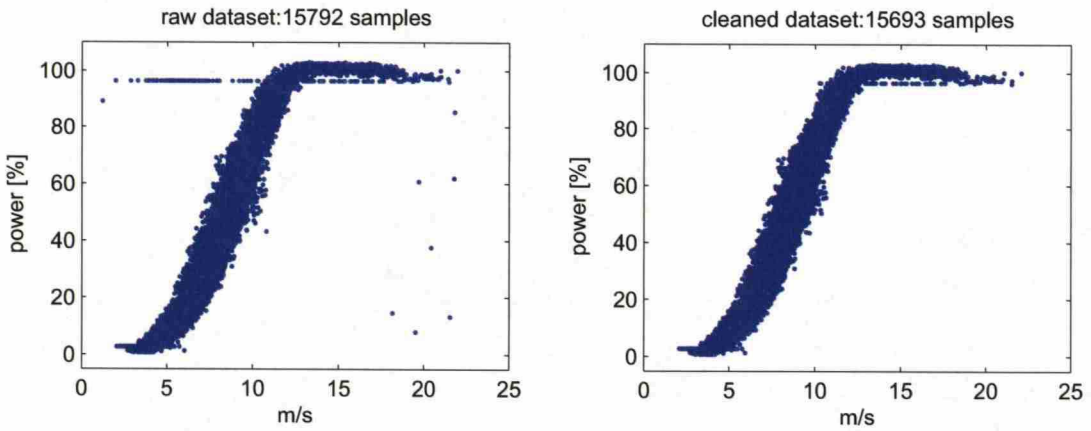

Figure 41: Generator power vs. wind speed in the raw measurement data (left) and the final reference dataset (right).

In this Chapter two potential icing incidents are picked from the data to illustrate the performance and accuracy of different methods and to highlight possible issues these methods might have in detecting the icing incidents.

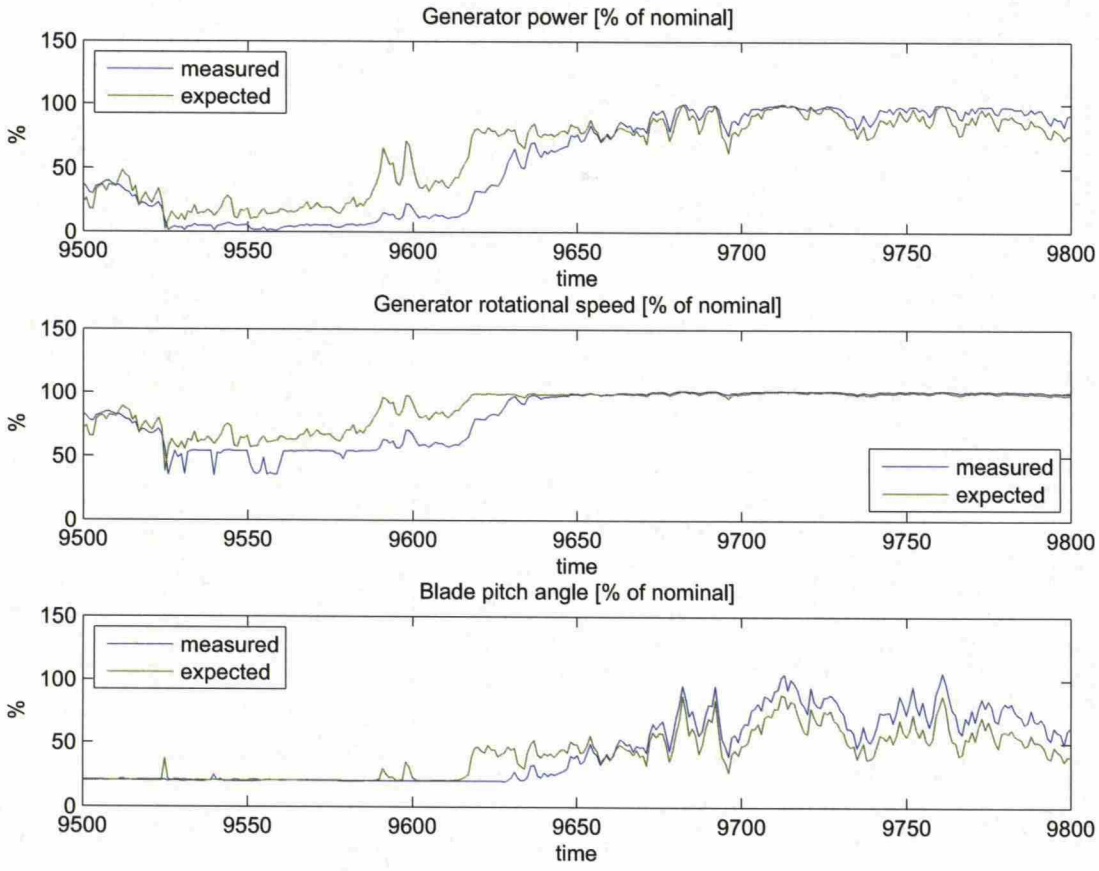

Figure 42: Potential icing incident in test data

6.1 Icing incident

An example incident that was detected by several methods is illustrated in figure 42. The behaviour of process variables is similar to predicted. When comparing to standard behaviour at the same wind speed generator power and generator speed and pitch angle have all dropped below their nominal values. The whole incident starts at time point 9530 and at the same time there is a very sharp drop in outside temperature (see figure 43). The event lasts approximately 20 hours. The event is marked at the beginning by a very obvious drop in temperature. The temperature itself is abnormally low, in general icing incidents are more common at temperatures closer to 0 ° C.

This event is visible when running the data through all the different methods described earlier. It causes a rather significant disruption in all control chart variables illustrated in figure 44. In all cases it takes several hours before any of the charts issue a warning. This is mostly related to the relatively light wind speeds at the moment of the incident. Once wind speed picks up, the control chart values finally cross the warning threshold. This is similar behaviour that was discovered in the simulated dataset as well. Detection does not work as reliably on lower wind

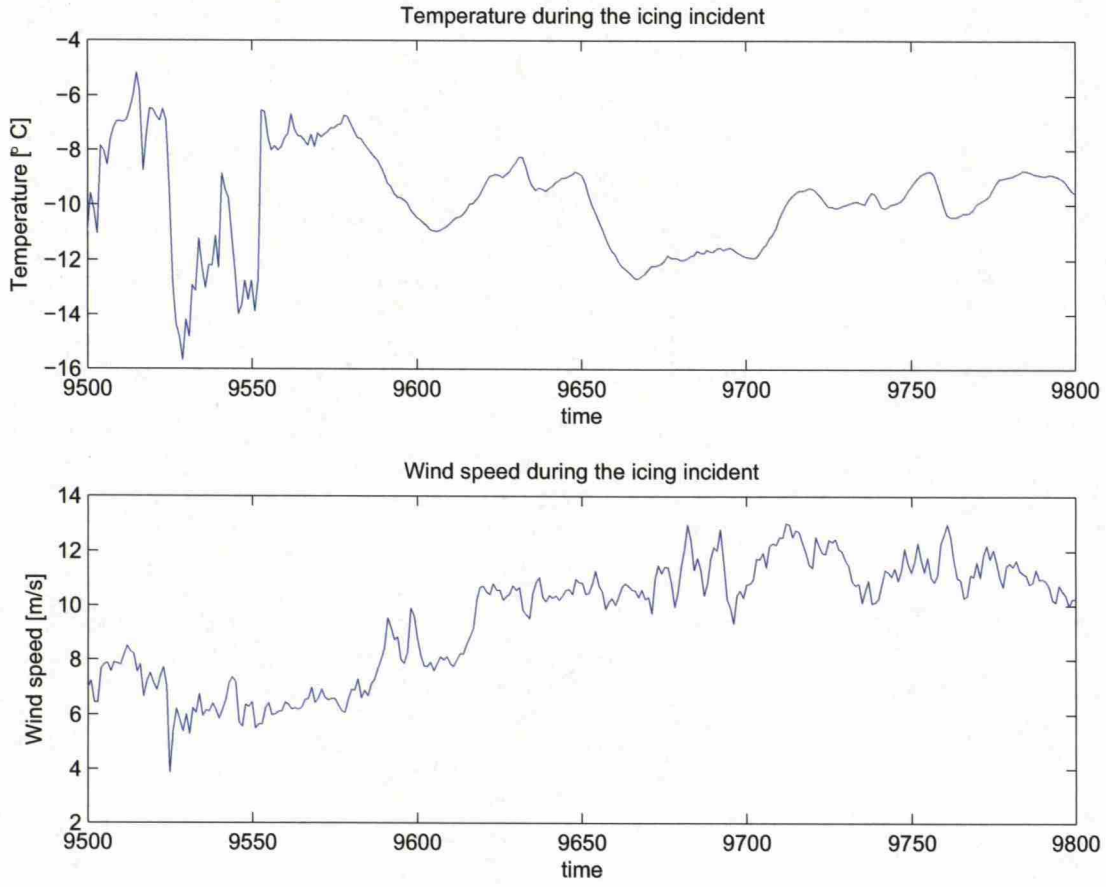

Figure 43: Temperature and wind speed during the incident in figure 42

speeds.

The recovery period is longer for CUSUM type charts, both of them come down to normal levels very slowly. The recovery time is almost as long as the length of the abnormality in process values. The recovery time is long, but the chart does return to normal levels eventually. Nevertheless, the return time is multiple hours in real time.

The PCA output for the same incident is illustrated in figure 45. Here the top two graphs display the values of PCA control charts and the bottom chart displays the final alarm output. In this case the incident can be assumed to start at 9520 and end at around 9660. This means that the PCA method is quite slow, the first alarm is issued at around 9590, and the recovery period is significantly slower than it is for most control charts. Once again, the slowness of initial detection might be related to small overall values of the process variables during the early stages of the icing incident.

The available data has a ten minute sample rate. Partly because of this the detection is significantly slower here than it was in the simulation study. There are several factors that can explain this disparity. First, the sample rate in the case study measurement data is significantly lower than the sample rate in the simulation

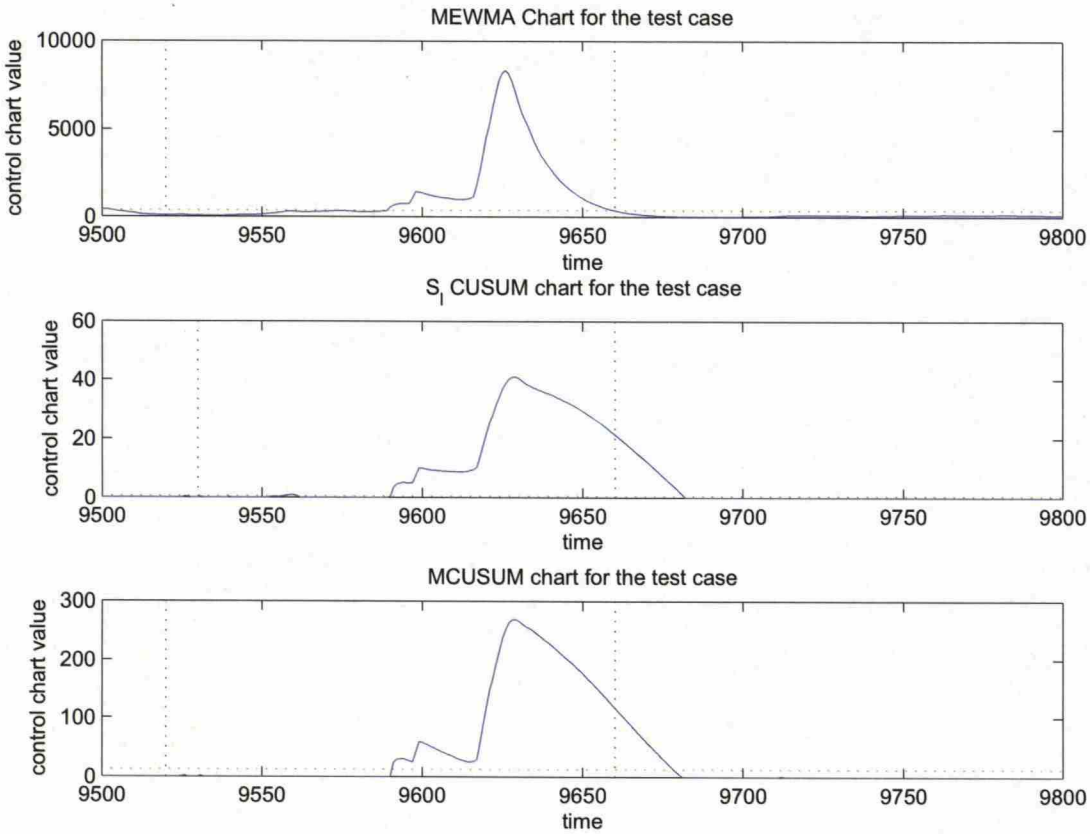

Figure 44: Control charts for the icing incident. Icing event starts at 9520 and ends at 9660.

study and ice detection seems to require a certain amount of measurements that contain an anomaly to be collected before an alarm is raised. As a result collecting the required amount of anomalous measurements after the icing event has started will take longer with the data used in the simulation. This alone does not explain the difference observed here. Most likely explanation is that ice accretion occurs relatively slowly and changes between concurrent measurements do not change very much. In the simulation the change was instant, which results in an unrealistically short detection time.

Noticeable in Fig. 45 is that the anomaly is seen a lot sooner in the F-test chart which relates to the T^2 statistic defined earlier. Issuing an alarm when either one of the test statistics is over the line would produce an alarm a lot quicker in this particular case. On the other hand it would also increase false positive alarms rather significantly.

The k nearest neighbours method detects icing roughly at the same speed as PCA, The end of the incident is also detected pretty quickly. Downside to this is the somewhat larger number of false positives. One way to address this is to issue an alarm only after n consecutive distance measurements are larger than the alarm

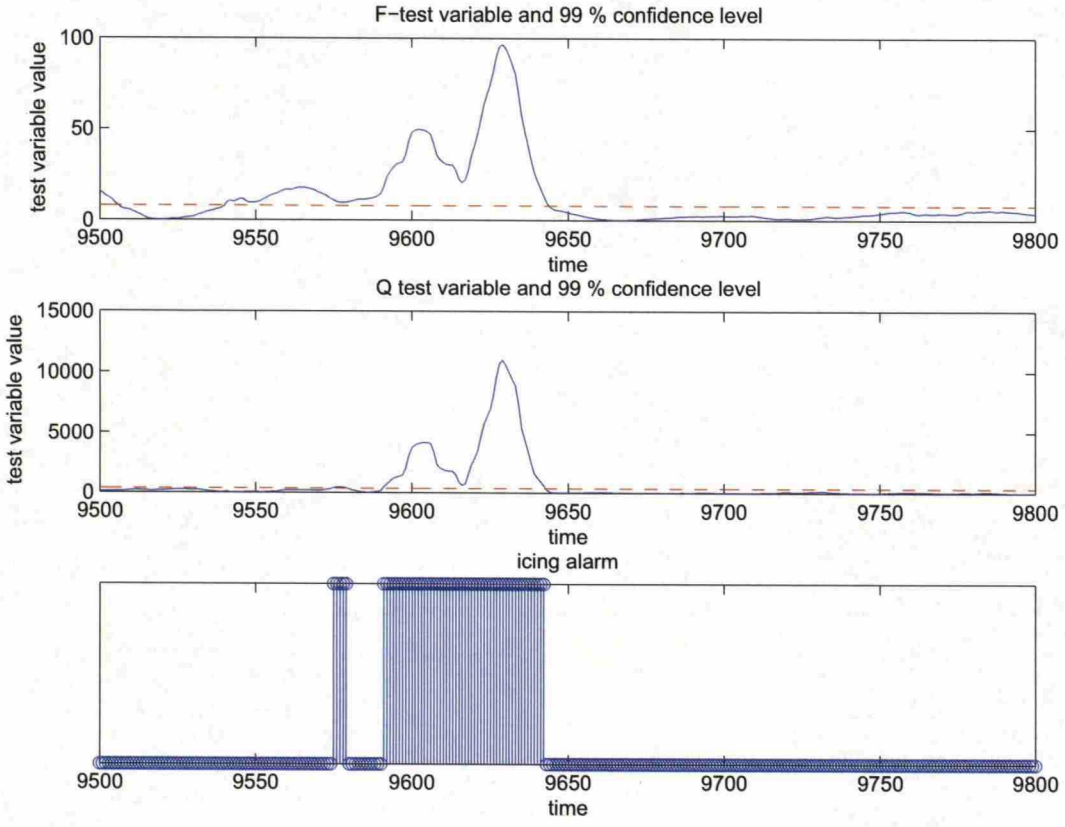

Figure 45: PCA method output for potential icing incident in test data

limit. Here $n > 1$, the exact value depends on the noisiness of data and the quality of measurements. This makes detection slower (delay is at least n) but it should decrease the number of false positive signals given by the method.

The detection happens rather long time after the initial drop in the process variable values becomes visible in Fig. 42. This can be explained with the relatively low wind speed at the time of the icing incident. There is less data i.e. fewer variables available from the real incident than there was for the simulations, which makes detection harder. Because of this the icing becomes visible through different methods only at a point where wind speed starts to grow, and the drop in values becomes more evident.

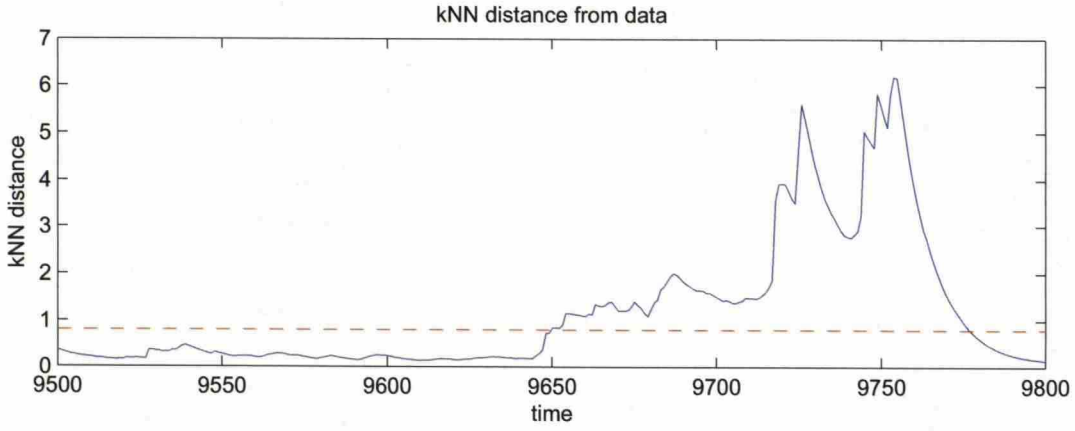

Figure 46: kNN method output for potential icing incident in test data

6.2 Comparison to ice sensor

For a part of the measurement period, there was an ice sensor located on the top of the nacelle. This makes it possible to compare the results of methods discussed here and the output of a state of the art ice sensor. The ice sensor does not indicate whether or not there is ice on the turbine blades, it only indicates that the conditions are right for icing. The ice sensor here measures signal strength in a wire. Ice accretion causes the signal to weaken. An alarm is issued if the signal strength drops below a predetermined level.

The ice sensor data can be plotted on the same timeline along with various control charts and other process data based indicators. There does not seem to be a clear-cut incident in this data set where all of the methods described in this work would issue an alert at the same time the ice sensor signal drops to a low level. Also these incidents do not really go hand in hand with anomalies in process variable values like they do in incident illustrated in Fig. 42. This emphasises the difficulty in actual icing detection and the need for a more reliable methods.

One incident is pictured in Fig. 47 and 48. The drop in ice sensor signal level is an indicator that the weather conditions were suitable for icing. There is a barely visible drop in process variable values in Fig. 48. The drop could be interpreted as being caused by icing, the conditions were suitable, and the resulting shift is to proper direction. The drop is relatively small, which means that it might as well be a part of normal variance in data and probably would not be noticeable without the ice sensor pointing towards it. There is also a smaller drop around the time point 1000, where the ice sensor also reacts. There the drop in ice sensor signal values is slightly smaller, but very clear regardless.

The ice sensor signal looks the way it does due to the operating principle of the sensor. When the sensor is iced badly enough that the signal level drops below a warning level, the sensor probe is heated until the signal returns to normal level i.e. until all the ice melts. After the ice is melted the heating is switched off. The ice

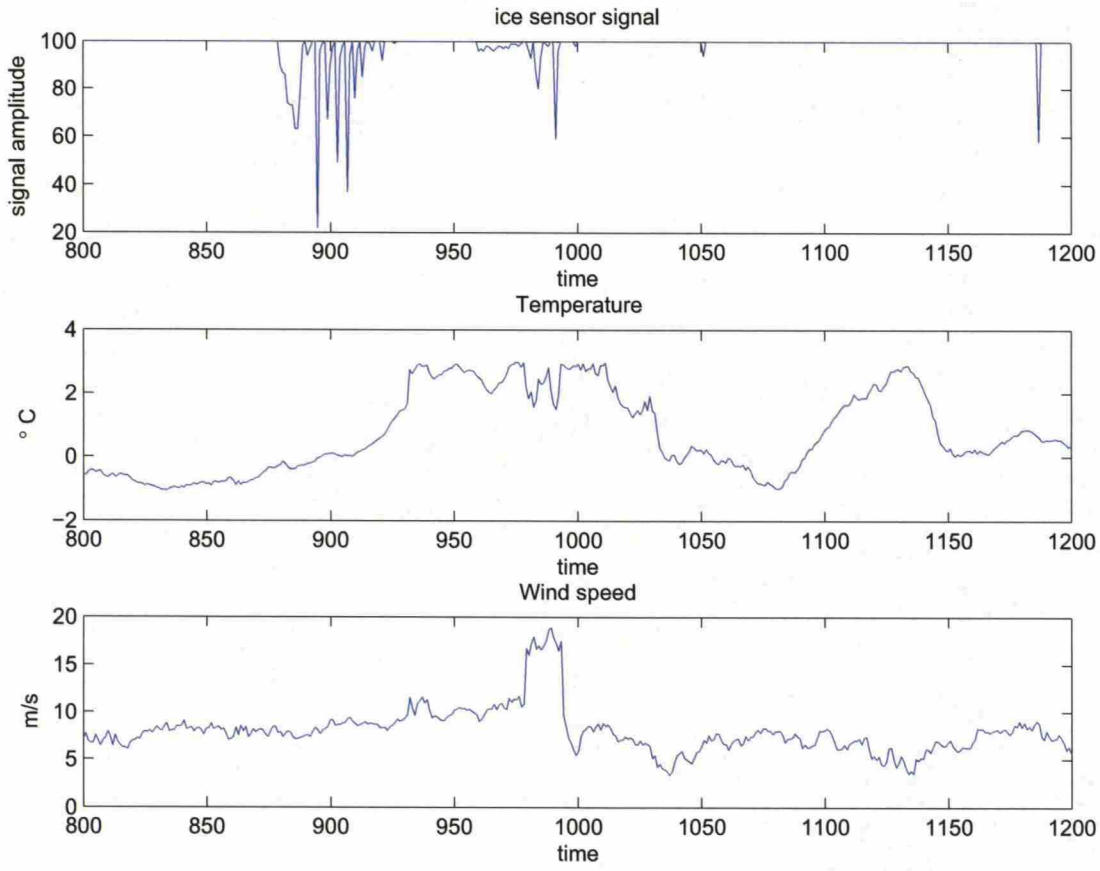

Figure 47: Ice sensor output, temperature and wind speed for a time period in test data

needs to be frequently removed from the sensor in order to gauge if the conditions continue to remain hostile or not. It is after all possible for the temperature to remain below zero without a significant icing risk. This heating cycle causes the ice sensor output signal to fluctuate heavily under icing conditions.

All the control charts in Fig. 49 do not cause a warning at both times the ice sensor reacts, the S_l CUSUM chart catches both but MCUSUM chart reacts only to the latter event. And the MEWMA chart does not react to either one. The common feature between the MEWMA and MCUSUM charts is that both of these use the covariance of variables in updating the distance measure. This might explain their similar behaviour. interestingly turbine power is significantly smaller during the latter incident, but for some reason the first one does not register in all control charts. There is an increase in the values of the MEWMA chart, but not large enough to raise an alarm. Granted the alarm limit used for MEWMA chart is rather arbitrary, but the value of the chart is rather small at this point.

The kNN chart displays that both of the potential incidents are visible as blips in the distance measure, but only the latter incident causes an alarm. Here the alarm limit is set at 95 % and the reference data cut off limit is set at 90 %. The

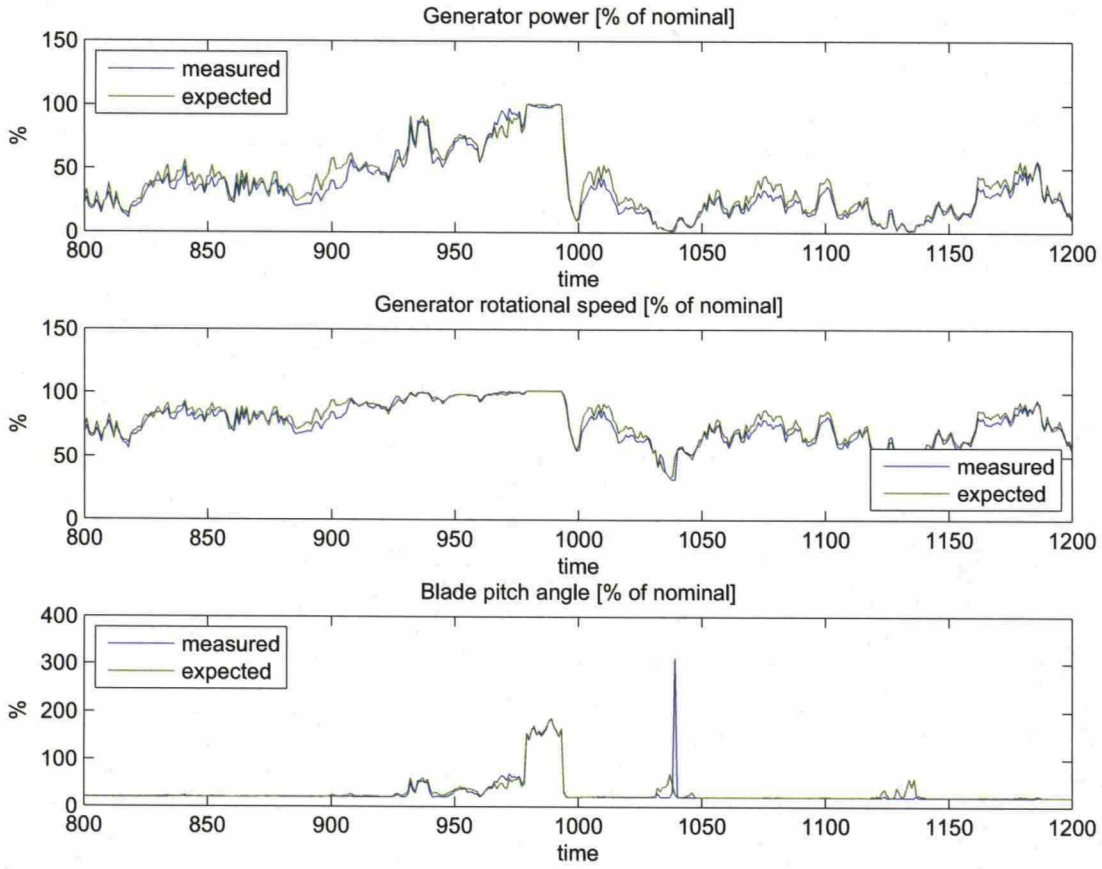

Figure 48: Process variable behaviour for time period corresponding to figure 47

measurements that do cause an alarm are therefore further away from reference data than 95 % of the cleaned reference data.

The PCA-based method does not react to the first incident at all, even though the ice sensor signal sees a rather massive drop here. It is likely that the relative drop in the process variables is not large enough for the PCA-method to pick up. It is visible in the F-test variable chart, but the Q-test variable is a flat line at this moment in time. Adjusting the number of used principal components does not seem to help here. The latter incident is caught And there is an additional blip later on at 1240 or so. Primary reason why the latter incident is detected is probably the sharp spike in the blade pitch angle measurement visible in Fig. 48. This kind of sudden and relatively large anomaly is always easiest to detect.

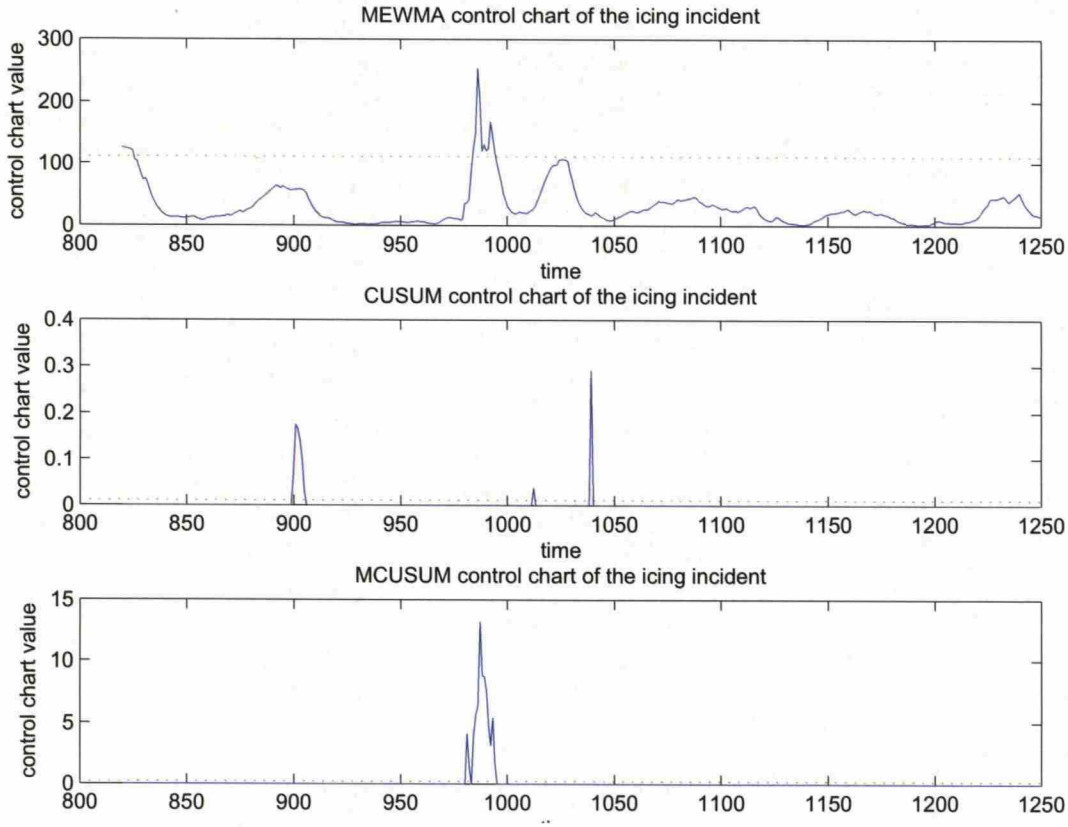

Figure 49: control charts for time period corresponding to figure 47

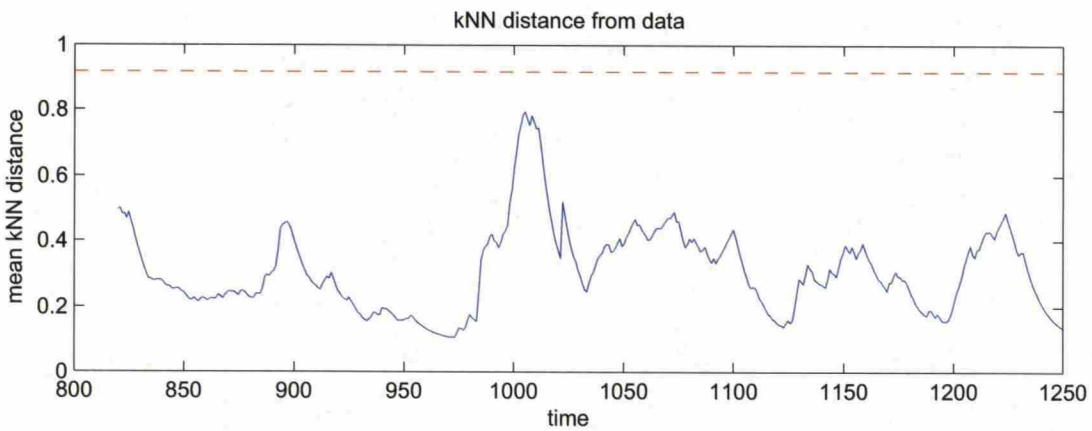

Figure 50: kNN method output for time period corresponding to figure 47

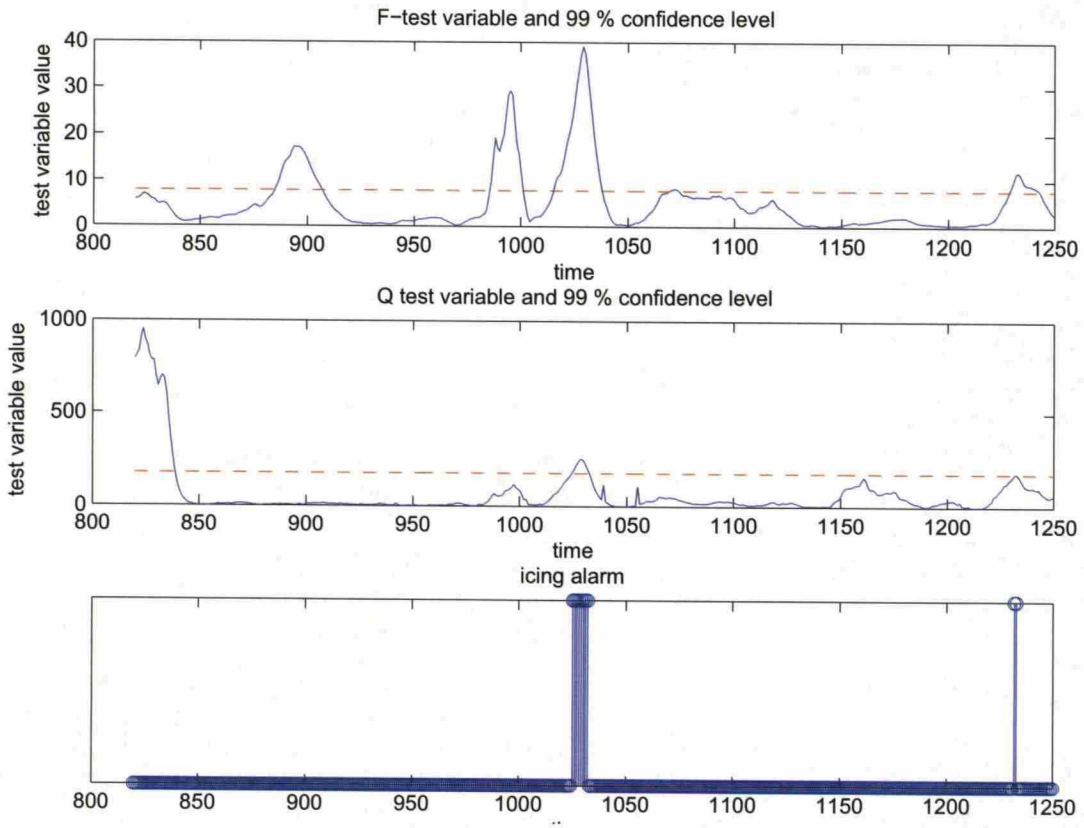

Figure 51: PCA output for time period corresponding to figure 47

6.3 Inclusion of ice sensor in to input data

A more refined solution would be to include the ice sensor into the data. This would add an additional measurement that is very likely to correlate with actual icing. This could increase detection accuracy and stabilise the system. As is clearly visible from Figs. 52 and 53, this is indeed the case. The effect of the ice sensor is so severe to the CUSUM charts, that the values of charts stay above the warning level the entire time between the incidents. This can happen in cases where there is a very large anomaly, the valuers of control charts might not return to a neutral level very quickly. This is a problem that can be triggered by other incidents as well. For example, if the turbine is forced to stop even though it is windy (because of a fault or other reasons), CUSUM control charts will jump to a very large value and it they might remain above the warning level for hours even after the turbine starts to operate normally.

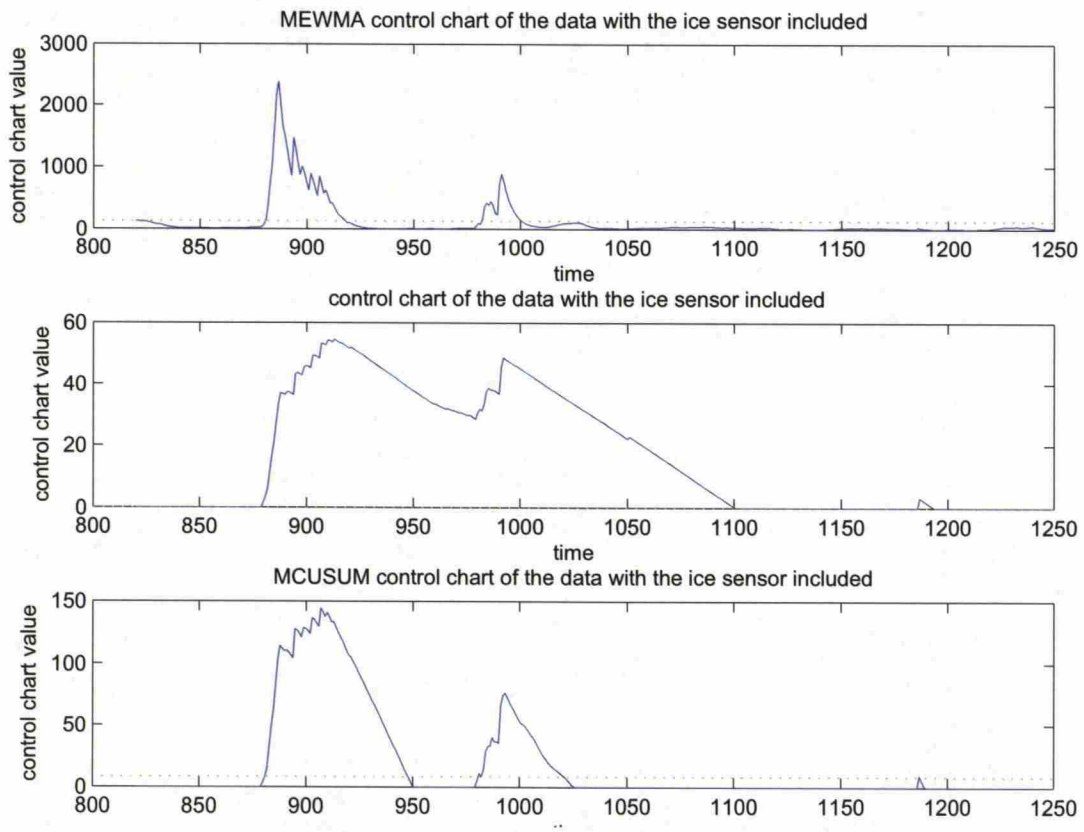

Figure 52: control charts for time period corresponding to figure 47 with the ice sensor included in the data.

Here the control charts were calculated using the same parameters as earlier and the result is that both the incidents where the ice sensor reacted are now clearly visible in all these control charts. In the same way PCA actually causes an alarm in both of these cases. The question here if this change in control chart behaviour really

is desirable or is it simply reacting to changes in ice sensor values. The shape of the ice detector sensor actually helps here as well. Because the ice detector has to melt the ice accreted on the sensor surface before it can make an updated measurement, the sensor signal resets to 100 often during a lengthy icing incident. As a result the variance of the ice sensor signal increases significantly during an icing incident. This in turn is easier to detect as an anomaly.

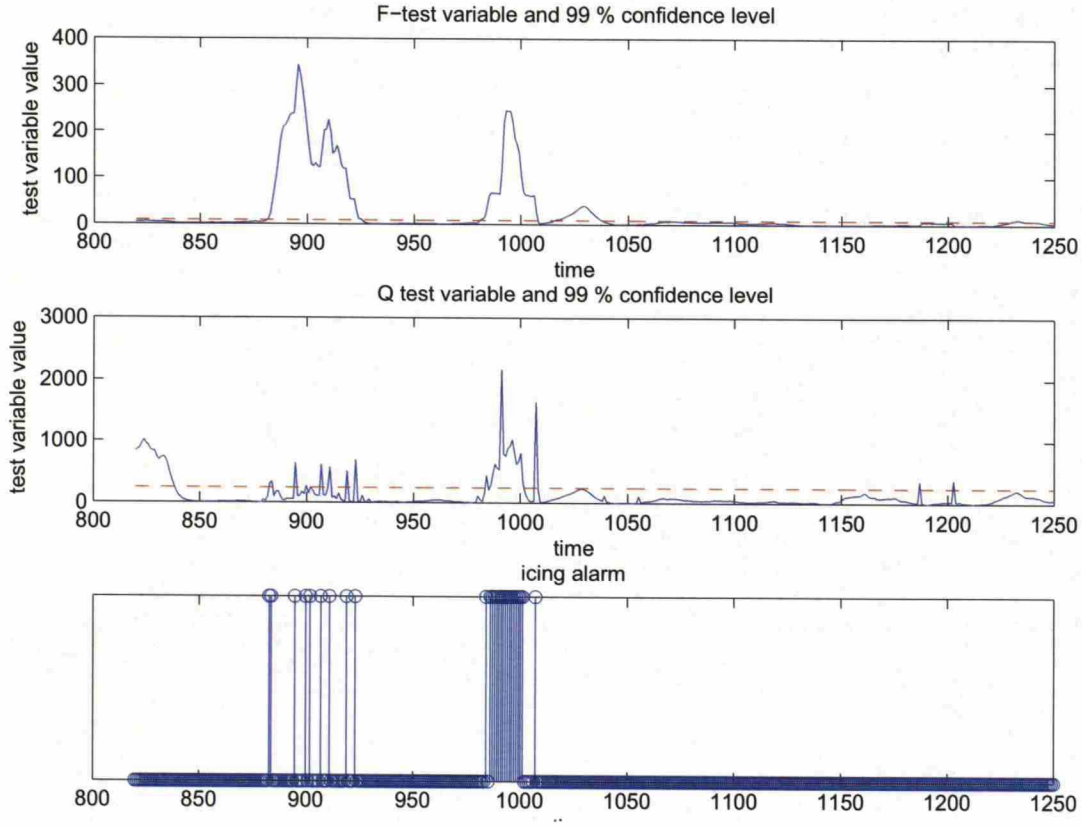

Figure 53: PCA output for time period corresponding to figure 47, with the ice sensor included.

The kNN method behaves in similar fashion as the others. Adding the ice sensor measurement on the side helped make the first potential icing incident visible here as well. The warnings happen pretty much simultaneously with the changes in the ice sensor signal values. It really looks like the ice sensor will dominate other measurements if it is added to the dataset, the change from signal in Fig. 50 to the signal in Fig. 54 is rather dramatic, even though only difference between the two cases is the addition of the ice sensor signal to the dataset. There is also a third spike in the kNN signal that was visible in an icesensorless run of the PCA-method (see Fig. 51). This does correspond to a small dip in process variables (visible at around time step 1180 in Fig. 48) but curiously was not visible until the inclusion of the ice sensor in to the dataset.

Generally adding the ice sensor to the process data can be useful and it will

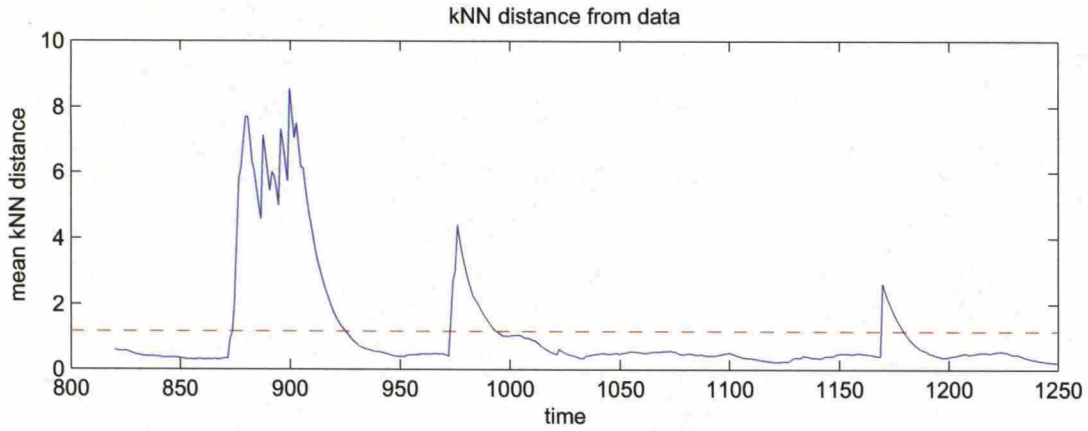

Figure 54: kNN output for time period corresponding to figure 47, with the ice sensor included.

probably increase detection accuracy. There might be issues relating to sample rates depending on how the ice sensor is implemented. It is after all a measurement that correlates to some degree with icing incidents, even if it is not foolproof. Depending on the implementation of the ice sensor, the ice sensor might start to dominate the output over other process variables. The sensor used here most definitely seems to improve results a lot. But at least the CUSUM control charts and the kNN distance measure seem to be

On the other hand using temperature as an additional measurement would not work. The basic principle behind all methods described in this work is to build a reference dataset during times when temperature is at a safe value, clearly above temperatures where icing might be possible. This means that temperature values between the two datasets will always be clearly different. Adding temperature would not produce any additional information about the state of the turbine.

6.4 Conclusions from the case study

The most obvious thing that was visible from the tests with real data is the sheer number of alarms. The accuracy of detection using only the data available here is not up to par. It is possible to dig out potential icing cases out of the data, but it is not a process that could be easily automated. Having more measurements could help with the accuracy, especially load measurements would be helpful. The load measurements could help because the changes in loads happen to the opposite direction when comparing to generator variables and blade angles. Methods discussed here do produce results, but are often triggered by normal variability in the system.

The data required quite a lot of work in order to get it into shape that was useful for testing these methods. There were pretty obvious faults in the system that were probably not caused by icing but at the same time caused huge issues for the detection methods. An example of this is illustrated in Fig. 55.

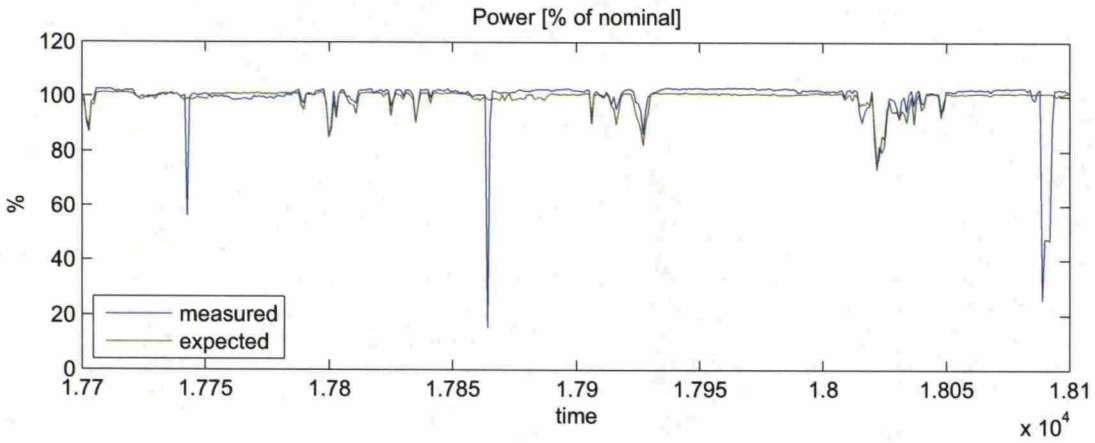

Figure 55: A series of faults in the turbine system

The sudden drops in turbine power are caused by some kind of fault in the system, that causes the turbine to shut down momentarily. Theses spikes in the data cause massive spikes in the control charts. It will also take a long time for the control charts to return to normal levels, because the change is so large and so quick. These spikes not only cause false alarms, but these false alarms can easily stay on for hours at time, even if there is nothing out of the ordinary happening in the system. Because faults cause such huge problems to control charts, knowing the turbine state is beneficial when using the methods described here. Knowing the turbine state would help ignore faults detected by the turbine control system and it would help ignore the large spikes in control charts these sudden changes in turbine values cause. Easiest way to cope with faults would be to suspend detection completely for the duration of a fault.

Putting all the pieces together, these methods do seem to work in a controlled environment, but the separation of icing from other events needs work. Currently these methods can act as a guideline or a starting point for search of icing events, but results are not conclusive and manual work is still needed in identifying and analysing interesting parts of the data. The methods do not reliably flag the events just yet, additional work is needed in order to increase detection accuracy.

Curiously enough the differences between the methods when analysing real data differ from the simulation study. Especially in the first icing incident PCA performance is comparable to all the control charts: all methods produce an alarm at roughly the same time. In all cases the different CUSUM charts also seem to also perform at a level comparable to MEWMA chart or better even in cases where the difference between the reference and the measurement is pretty small. Granted, all the cases studied do appear to reach the same wind speeds that seemed to work best in the simulation study as well.

In the first case shown here the kNN-based method reacted a lot slower than all others, and in the second case only reacted mildly once. The surprising slowness in the first example case is harder to explain what causes this slowness in this particular

case since other methods seem to work and the detection speeds of kNN seem to be line with others in the latter test case.

The differences between different methods seem to be less predictable when using real data. The produced results seem to vary from case to case. A solution to this could be to run several methods side-by-side and issue an alarm only when multiple methods give out a warning. For different control charts this could be an easily implementable solution.

A big problem in analysing the effectiveness of the methods discussed here is the fact that the data used is collected blindly, there are no verified icing events in the test data set. There are events that look promising and there are warnings issued by the ice sensor, but these are all just likely events, there is no real certainty in this.

7 Conclusions and future work

The results here show that the methods discussed work very reliably on controlled data. The results of the simulation study are very good and all the methods introduced do produce good results there. The detection accuracy of most methods seems to be related to the wind speed, with the best results coming from the medium wind test case, where wind speeds of the iced part of the data mostly stay below the rated wind speed of the simulation model. Detection does not work as well at lower wind speeds mostly due to the fact that all the variables observed do not differ from the reference at the wind speeds used in the low wind test case (see blade root moment in Fig. 22).

With the simulation data, across all datasets, the exponentially weighted moving average control chart performed the best. Additional good feature of the MEWMA control chart is the fact that it seemed to perform well without requiring wind speed specific tuning unlike other control charts.

When operating on real data, the control charts seem to produce better results as well, even though the difference is not as clear. Evaluation of performance in the case study is somewhat difficult, because of a lack of certain ice cases in the data. The most obvious looking incidents are caught by all methods but they all produce some false positive detections and some potential icing incidents are only caught by some of the methods.

With the real data, the control charts seem to be the best approach as well. Here the difference between MEWMA and different CUSUM charts is not as clear. This might be related to the fact that there were more measurements available in the simulation study: there were no load measurements included in the case study. The difference between detection accuracy might not be as evident because of this, after all the loads are often the variables that make the biggest difference in detection accuracy.

The excessive number of false positives or suspicious detections in the real data test case makes it hard to declare these methods useful just yet. There is still work that needs to be done in order to improve the overall accuracy of these methods. This could mean adding measurements or adding different statistics into the input data. Right now the only data used is the mean values from a time series. It might be smarter to incorporate some higher order statistics into the methods as well. For example studying the behaviour of variable standard deviations might produce interesting results. On the other hand for strongly fluctuating signal like accelerations or loads the arithmetic mean might not be the best way to represent the signal value. Results might be better if the methods were modified to look at changes in RMS values.

The behaviour of fatigue loads in an iced turbine needs to be examined outside of a simulation. This could perhaps be the next logical step if suitable dataset is found.

Another pressing concern in the quality of input data. There is a very real demand for some kind of filter for input data so that obvious outliers generated by measurement errors or faults in the system can be removed from the data first,

before affecting the detection results.

Also wind turbine behaviour is always affected by its surroundings. Buildings, vegetation, landscape and most importantly other wind turbines have a noticeable effect on turbine performance. Wind direction needs to be taken into account somehow in order to get a detection method that works in a real world environment as well.

References

[1] Munteanu, I., Bratcu, A. I., Cutululis, N., Ceang, E., Optimal Control of Wind Energy Systems: Towards a Global Approach. Springer, London, 2008

[2] Bianchi, F.D., De Battista, H., Mantz, R.J., Wind Turbine Control Systems, Principles, Modelling and gain Scheduling Design, Springer-Verlag 2007

[3] Homola, M., Nicklasson, P., Sundsbø, P. Ice sensors for wind turbines, Cold Regions Science and Technology 46 (2006) 125–131

[4] Parent, O., Ilinca, A., Anti-icing and de-icing techniques for wind turbines: Critical review, Cold Regions Science and Technology 65 (2011) 88–96

[5] Wang, X., Bibeau, E., Naterer, G., Experimental Investigation of Energy Losses due to Icing of a Wind Turbine. Challenges of Power Engineering and Environment, pp. 1143–1147, Springer-Verlag Berlin Heidelberg 2007

[6] Hochart, C., Fortin, G., Perron, J., Wind Turbine Performance under Icing Conditions, Wind Energy 2008; 11:319–333, DOI: 10.1002/we.258

[7] Barber, S., Wang, Y., Chokani N., Abhari R. S., The Effect of Ice Shapes on Wind Turbine Performance, IWAIS XIII, Andermatt, September 8 to 11, 2009

[8] Jonkman, J.M, Buhl Jr., M.L., FAST User's Guide Technical Report NREL/EL-500-38230 2005

[9] Manjock, A. Evaluation Report: Design Codes FAST and ADAMS® for Load Calculations of Onshore Wind Turbines. Report No. 72042. Humburg Germany: Germanischer Lloyd WindEnergie GmbH, May 26, 2005.

[10] Jonkman, J., Butterfield, S., Musial, W., Scott G. Definition of a 5-MW Reference Wind Turbine for Offshore System Development, Technical Report NREL/TP-500-38060 February 2009

[11] Finnish Meteorological Institute, VTT Finnish Icing Atlas, http://www.tuuliatlas.fi/icingatlas/index.html, Cited 6.11.2012

[12] Frohboese, P., Anders, A., Effects of Icing on Wind Turbine Fatigue Loads, J. Phys.: Conf. Ser. 75 012061 (2007) doi: 10.1088/1742–6596/75/1/012061

[13] Ganander, H., Ronsten, G., Design load aspects due to ice loading on wind turbine blades BOREAS VI9 - 11 April 2003 Pyhätunturi, Finland

[14] Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri S., A review of process fault detection and diagnosis Part I: Quantitative model-based methods, Computers and Chemical Engineering 27 (2003) 293–311

[15] Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri S., A review of process fault detection and diagnosis Part II: Qualitative models and search strategies, Computers and Chemical Engineering 27 (2003) 313–326

[16] Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri S., A review of process fault detection and diagnosis Part III: Process history based methods, Computers and Chemical Engineering 27 (2003) 327–346

[17] Bin Shams, M.A., Budman, H.M., Duever, T.A., Fault Detection, identification and diagnosis using CUSUM based PCA. Chemical Engineering Science 66 (2011) 4488–4498, doi:10.1016/j.ces.2011.05.028

[18] Isermann, R., Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, Springer-Verlag Berlin Heidelberg 2006

[19] Liu, J. Data-driven fault detection and isolation for multimode processes, Asia-Pac. J. Chem. Eng. 2011; 6: 470–483 DOI:10.1002/apj.549

[20] Daszykowski, M., Kaczmarek K., Vander Hayden Y., Walczak B., Robust Statistics in data analysis - A review Basic concepts Chemometrics and Intelligent Laboratory Systems 85 (2007) 203–219

[21] Li, C., Jeng, J., Multiple sensor fault diagnosis for dynamic processes, ISA Transactions 49 (2010) 415–432

[22] Thornhill, N., Locating the source of disturbance, Process Control Performance Assessment, Advances in Industrial Control, 2007, 199–225, DOI: 10.1007/978–1–84628–624–7_6

[23] Camci, F., Chinnam, R. B., Ellis, R. D., Robust kernel distance multivariate control chart using support vector principles, International Journal of Production Research (2008), 46:18, 5075-5095

[24] Timusk, M., Lipsett, M., Mechefske, C., Fault detection using transient machine signals, Mechanical Systems and Signal Processing 22 (2008) 1724-1749, doi:10.1016/j.ymssp.2008.01.013

[25] Markou, M., Singh, S., Novelty detection: a review – part 1: statistical approaches, Signal Processing 83 (2003) 2481–2497, doi:10.1016/j.sigpro.2003.07.018

[26] IEC-61400-12-1: Power Performance measurements of electricity producing wind turbines, IEC 2005

[27] Vieringa, J. E. Statistical Process Control of Serially Correlated Data, PhD thesis, University of Groeningen, Department of Econometrics, 1999

[28] Yeah, A., Huwang, L., Wu, C-W., A multivariate EWMA control chart for monitoring process variability with individual observations, IIE Transactions (2005) 37, 1023–1035 DOI: 10.1080/07408170500232263

[29] Hawkins, D. M., Choi, S., Lee, S. A General Multivariate Exponentially Weighted Moving-Average Control Chart, Journal of Quality Technology; Apr 2007; 39, 2; ABI/INFORM Complete pg. 118

[30] Prabhu, S. S., Runger, G. C., Designing a multivariate EWMA control chart. Journal of Quality Technology; Jan 1997;

[31] Stefatos, G. Ben Hamze, A. Dynamic independent component analysis approach for fault detection and diagnosis, Expert Systems with Applications 37 (2010) 8606–8717

[32] Bersimis, S. Psarakis, S., Panaretors,, J. Multivariate Statistical Process Control Charts: an Overview, Qual. Reliab. Engng. Int. 2007; 23:517–543

[33] He, Q., Yan, R., Kong, F., Du, R., Machine condition monitoring using principal component representations, Mechanical Systems and Signal Processing 23 (2009) 446–466, doi:10.1016/j.ymssp.2008.03.010

[34] MacGregor, J. F., Kuorti, T. Statistical Process Control of Multivariate Processes, Control Eng. Practice, VoL 3, No. 3, pp. 403-414

[35] Yoon, S. MacCregor, J. F. Fault diagnosis with multivariate statistical models part I: using steady state fault signatures, Journal of Process Control 11 (2001) 387-400

[36] Kusiak, A., Zheng, H., Song, Z. Models for monitoring wind farm power Renewable Energy 34 (2009) 583–590

[37] He, Q., Wang, J. Fault Detection Using the k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes, IEEE transactions on semiconductor manufacturing, vol. 20, no. 4, November 2007

[38] Hodge, V.J., Austin, J. A survey of outlier detection methodologies. Artificial Intelligence Review, 22 (2). (2004) pp. 85-126.

[39] Friedman, J.H., Baskett, F., Shustek, L.J., , An Algorithm for Finding Nearest Neighbors, Computers, IEEE Transactions on , vol.C-24, no.10, pp. 1000- 1006, Oct. 1975 doi: 10.1109/T-C.1975.224110