UNIVERSITY OF AGDER

Using Hidden Markov Models for fault diagnostics and prognostics in Condition Based Maintenance systems

By
Magnus Bjerkeseth

Thesis submitted in Partial Fulfillment of the
Requirements for the Degree Master of Technology in
Information and Communication Technology

Faculty of Engineering and Science
University of Agder

Grimstad
May 2010

Abstract

Condition Based Maintenance (CBM)) is a concept that has become more and more important as the cost, size, and complexity of mechanical components has increased. As more sensor equipment has become available it has become possible to measure different kinds of status data, such as vibration, temperature, or electric current, from different kinds of mechanical components. By using this status data it should be possible to determine the health of mechanical components, and determine when they need maintenance, and what parts needs to be replaced.

Hidden Markov models (HMM) is a statistical model for modelling systems that evolve through a finite number of states. By using HMMs it is possible to detect and recognize different kinds of anomalies and errors present in the system. It is also possible to estimate when the system is going to be in a state where it can not be expected that the system is going to function properly (error state).

During this thesis I have examined several different ways of applying HMMs to tasks related to condition based maintenance. HMMs have been tested in their use in anomaly detection, current state detection, and future state prediction. All these tasks have been performed with little available training data to demonstrate how HMMs can still be used even if little prior knowledge about the system is available.

Tests have been performed to evaluate how well HMMs can detect unknown data (anomaly detection), determine the current health of the system (current state detection), and predict the future health of the system (future state prediction). During all the tests it has also been a focus on how limited training data and expert knowledge about the system will influence the approach to the problem and the results of the tests.

Out of all the test that were performed anomaly detection is the task that was least influenced by the lack of training data or prior knowledge. Current state detection accuracy was most influenced by the lack of prior knowledge and the quality of the training data, while future state prediction could benefit from more training data.

Based on the test it is possible to say that anomaly detection can be performed with just a minimum of training data. However, prior knowledge about the system becomes more important because anomalies needs to be classified. Current state detection requires more training data than anomaly detection. Prior knowledge about the system is also important when training HMMs to recognize error states. Future state prediction requires a lot of varied training data to be able to perform reasonably accurate predictions for new components, that may have a longer life-cycle than previously seen components.

The results of these tests gives an indication about the requirements for training data, and knowledge about the system where anomaly detection, current state detection, and future state prediction is being performed.

Preface

This master thesis is submitted in partial fullfillment of the requirements for the degree Master of Science in Information and Communication Technology at the University of Agder, Faculty of Engineering and Science in Grimstad, Norway. This work was carried out under the supervision of Ole-Christoffer Granmo at the University of Agder, and Pål Berg at Origo-Mobikom.

I would like to thank my supervisors who has provided advice and input through this project. Secondly I would like to thank my family who have supported and encouraged me throughout this project, and the rest of my time attending the University of Agder.

Magnus Bjerkeseth Grimstad, May 2010

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the following sections we will briefly introduce condition based maintenance (CBM), and previous work that relates to CBM, hidden Markov models and the problems examined during this thesis. Further we will present the full problem definition, details of all problems that will be examined, and how the problems are related to previous work. At the end of this chapter there will be a brief outline of the rest of this thesis, that will provide a short summary of the content of each chapter.

1.1 Condition based maintenance and importance of research

The traditional techniques for system maintenance are preventive maintenance and corrective maintenance. Preventive maintenance takes the form of scheduled maintenance actions aimed at the prevention of breakdowns and failures. The primary goal of preventive maintenance is to prevent the failure of equipment before it actually occurs. It is designed to preserve and enhance equipment reliability by replacing worn components before they actually fail. [7]. Corrective maintenance is a form of maintenance which is performed after a fault emerges in a system, with the goal of restoring the system to a working condition. In some cases, it can be impossible to predict or prevent a failure, making corrective maintenance the only option. In other instances, a poorly maintained system can require repairs as a result of insufficient preventive maintenance, and in some situations people may chose to focus on corrective, rather than preventive, repairs as part of a maintenance strategy.[1].

However, both preventive and corrective maintenance have drawbacks that make them unsuitable in certain situations. For example, the regular maintenance scheduled with the preventive maintenance scheme does not take into account that components does not necessarily need maintenance just because they have been used for a long time. This may lead to scheduling maintenance and stopping the system even when it is not needed. Performing maintenance when it is not necessary leads to an increase in costs for new components, and reduced production time since the machinery can't be used when maintenance is being performed. In the case of corrective maintenance it might be possible to let cheap and unimportant machinery run until it fails, however, this is not an option with expensive and/or important machinery.

The ability to predict or estimate when components in a mechanical system are going to fail is important for affordable system operation. Condition based maintenance (CBM) means that maintenance is performed based on an assessment or prediction of the health of the components in a system. If it is possible to predict degradation of a component, it will be possible to schedule maintenance and order a replacement component before the component fails completely [3].

The idea of condition based maintenance (CBM) is to maintain equipment only when it is necessary. CBM uses measured status data (including, but not limited to, vibration, temperature, oil debris, or electric current) from the equipment to prioritize and decide on maintenance schedules. Such an approach may allow the system to be operational for longer periods of time, thereby increasing production time and reducing downtime due to maintenance. By having a good estimate about when the system is going to need maintenance it is possible to order replacement parts in good time before the scheduled maintenance takes place. This is especially useful when parts are too large, heavy, or expensive to keep in regular storage close to the facilities where maintenance is performed [3].

A CBM system will monitor the health of a mechanical component through measurements of different kinds of data that can say something about the health of the component. The measured data will be analysed to determine the current health of the component, and in some cases a prediction about the future health can be made. When creating a CBM system, expert knowledge about the system is necessary. Expert knowledge should come from someone who has extensive experience on working on the system and can give an indication on different component errors related to specific data measurements.

A hidden Markov model(HMM) is a statistical model that can be used to analyse data gathered from, among other things, mechanical components. The system that is modelled by a HMM is assumed to be a Markov process [16] with unobserved states [4]. A HMM can be trained to recognize data measurements from a mechanical component. This makes it possible to train a HMM to recognize the different health states of a mechanical component. A fully trained HMM can be seen as a state machine, where each state in the HMM represents a different health-state of the mechanical component. Transition between states is probabilistic, so if a component is observed to be in state 1 at time t it is possible to predict which state the component is most likely to be in at time t+1 and so on.

The states in the system being represented by a hidden Markov model are hidden to outside observers. It is only possible to know which state the system is in by the feedback the hidden Markov model returns when given sensor data. In Figure 1.1, Y1, Y2, Y3, and Y4 are vectors that represent data collected from different sensors, each vector may contain several different kinds of observations (for example vibration, temperature, and vibration from a different sensor). X1 (good), X2 (stable), and X3 (failing) represents the possible states the system can be in. The data recorded in Y1 might be taken when the component was freshly installed and will have a greater probability of indication that the component is in state X1. State X1 will indicate that the component is new, or as good as new, and has not suffered any damage. State X2 indicates that the component has suffered some wear and tear but will still continue to function properly. State X3 represents that the component has suffered significant damage, and will need to be replaced as soon as possible. The transition between states is probabilistic and represent how probable it is for a component to appear in a different state in the future.

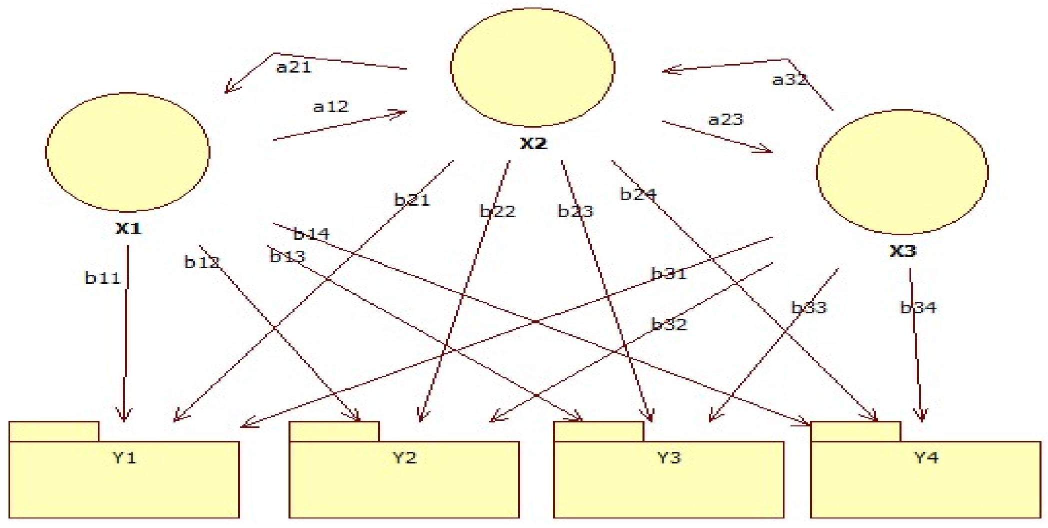

Figure 1.1: An example hidden Markov model where X = states, Y = input vectors, a = probability of transition between states and b = output probabilities.

1.2 Previous work

In this section, we will give a brief overview of some of the previous work that has been carried out in the field of condition based maintenance and hidden Markov models.

A paper released in 2005 [17] details how CBM can be used to detect faults and predict failures of bearings. This paper describes how hidden Markov models (HMM) can be used to determine the current health and the expected remaining life of a bearing. This is done by using vibration data collected from old bearings, and using this data to adjust the hidden Markov model so that it can give a good representation of a component life-cycle. Measurements of new bearings are then analyzed by the hidden Markov model, and in this manner the health of a component can be estimated. To predict the remaining useful life a degradation index is obtained from the HMM. This degradation index is then processed further to estimate the remaining useful life of the component.

An article released in 2009[11] uses hidden Markov model and analyses sensor readings on electromechanical actuators. An actuator is a mechanical device for moving or controlling a mechanism or system. An actuator typically is a mechanical device that takes energy, in this case electricity, and converts that into motion of some form (pushing, pulling, rotation). In this case a hidden Markov model is used simultaneously with other fault detection equipment. The hidden Markov model has been used to monitor and detect faults in the electric system that the other fault detection equipment was not able to detect.

Bunks, McCarthy and Al-Ani[6] goes into detail on how data and expert knowledge about a mechanical system can be used to train HMMs to detect faults and different kinds of defects in a system. Although future state prediction (fault development) is not performed, they indicate how the future state prediction might be performed by using a hidden Markov model and having expert knowledge about the system.

Baruah and Chinnam[5] describes how several hidden Markov models where created to detect the current state of a drilling system. The paper also describes a method which can be used to predict the future states of the system by estimating the points (in time) when the system will go from one state to another.

1.3 Problem definition

> "Condition Based Maintenance mainly uses measured status data (including, but not limited to, vibration, temperature, oil debris, or electric current) to determine the health of a mechanical system. When a large amount of data has been collected, it is also possible to use this data to predict the future health of the system. The purpose of this master thesis is to use Hidden Markov Models to attempt to classify the current condition (diagnosis) and make a prediction (prognosis) for the future condition of the mechanical system by using current and historical data."

During this project hidden Markov models will be used to accomplish three different tasks related to system diagnostics and prognostics. The tasks are anomaly detection, system state recognition, and future state prediction. The goal is to accomplish these three tasks by using the knowledge that can be distilled by using hidden Markov models.

Anomaly detection is going to examine how a hidden Markov model can be used to detect unknown measurements. System state recognition involves determining which health-state the system is currently in. Future state prediction involves estimating when the system is going to change states.

Another challenge faced in this thesis is the amount of available training data. Usually massive amounts of training data are not available when CBM systems are created, and the system has to be created with this limitation in mind. This is also true for expert knowledge about the data. Because of this, a part of the problem is to determine how much data and expert knowledge is required to perform the different tasks.

1.4 Contribution

CBM is a technique that has not yet been implemented on a large scale in industry. Many companies have installed various sensors on their equipment and used this gathered information to determine the current health of the system. However, this gathered information has seldom been used to determine the future health of the system.

During this project hidden Markov models will be used to detect unknown data (anomaly detection), determine the current state of the system, and estimate future state transitions.

If the hidden Markov model approach to CBM health analysis is successful then this project will help to show a viable method to CBM health analysis. However, if the hidden Markov model approach is unsuccessful then this project will hopefully be able to pinpoint the reason for failure, and suggest what steps must be taken to make sure that hidden Markov model can be used for health diagnosis and prognosis in CBM systems.

During this thesis it will be shown how anomaly detection, current state detection, and future state prediction is affected when only a small amount of data is used to train the different HMMs. This will hopefully help to underline the importance of having a large dataset when creating HMM based CBM systems.

1.5 Assumptions and limitations

During work on this thesis several hidden Markov models(HMM) was created. The HMMs was trained from data obtained from Origo Mobikom. The obtained data was temperature measurements from two electric motors. To the best of our knowledge the data measurements had been performed in real life situations (i.e. the electric motors were in regular use when the data was collected).

The nature of the electric motor data placed several constraints on it usage. The data measurements contained huge random leaps between measurements. All recorded temperature values were measured in 1 minute intervals. However, there were large sections of data missing, often gaps covering several days, months, or sometimes an entire year. Because of this it is not possible to get an overview of how the data changes when the motor gradually wears down over time. Having a good overview of an entire life-cycle is necessary for future state prediction, since this prediction needs information about all possible health-states the motor can go through.

There was also a lack of expert knowledge about the data. Without this knowledge we had no way of knowing about the states the data might describe. Instead we made a decision to classify the data into 2 possible health states. Expert knowledge might have changed this decision.

Even though there are some shortcomings in the electric motor data, it can still be used to perform anomaly detection and current state detection.

For demonstrating future state prediction, some generated data will be used. The amount of generated data is the minimum amount required to be able to perform a future state prediction. The reason for generating a minimum amount of data is that it represents a real life situation were someone wishing to construct a CBM system while using HMM might not be able to gather enough data be able to properly train the required HMMs or the state prediction system.

The limitation and assumptions are summarized as follows:

- Electric motor temperature data not suitable for performing future state prediction. This is countered by generated data.
- Lack of expert knowledge means that the defined states of the electric motor data is not as good as it could have been if expert knowledge had been available, and some of the results may suffer because of this. We will however try to offer other solutions that might improve the test results.
- Small amounts of training data for prediction. Although more varied data could be generated, the generated data can be used to show how prediction becomes affected when little training data is available.

1.6 Report outline

Chapter 2 contains a short introduction to hidden Markov models and the associated algorithms. This chapter also gives a short introduction to the types of hidden Markov models we will use in the rest of this thesis

Chapter 3 is about the approach that we took when we solved the problems. This chapter will detail the hidden Markov models that were created and how they can be used to solve the presented problem.

In chapter 4 we will present the results of the tests performed to evaluate whether the created hidden Markov models managed to perform anomaly detection, estimate current health, and predict the future health development of the system.

Chapter 5 is a summary of all the work, tests and results, as well as the conclusion of our work.

Chapter 2

Introduction to hidden Markov models

In this chapter we will give a brief introduction to hidden Markov Models. The introduction will include the two basic types of hidden Markov Models, as well as the three most important problems. At the end of this chapter we will inform about the type of hidden Markov model we used while working on this thesis.

2.1 Introduction to hidden Markov models

This section contains a short introduction to hidden Markov models. For more detailed information about hidden Markov models and the associated algorithms we suggest reading "A tutorial on hidden Markov models and selected applications in speech recognition" by Lawrence R. Rabiner [12].

2.1.1 HMM general

A hidden Markov model (HMM)) is a statistical model [4] for modelling systems that evolve through a finite number of states [5]. The states for the system that is being modelled is hidden to the observer. However, by letting a HMM analyze observations made from the system, it is possible to determine which state the system is most likely currently in.

A HMM will consist of several states (N), an initial probability value(π) for each state indicating how likely it is for a new input sequence to start in a given state, a transition probability matrix (A) indicating the likelihood of transitioning from one state to another, and an output probability distribution (B) that indicates how likely it is for a certain measure value to come from a given state. The sum of all start probabilities must be equal to 1, and the sum of all all elements in a row in the transition probability matrix must also be equal to 1. The simple notation for a HMM is λ = (A, B, π) [12].

There are two main types of hidden Markov models, discrete and continuous. The main difference between them lies in the input which the HMM is able to accept, and the way which the input is processed.

2.1.2 Discrete hidden Markov model

The first version is the discrete HMM. For this version there exists a limited number of observations which can be made. An example of a discrete HMM is shown in Figure 2.1. The HMM described in Figure 2.1 is supposed to estimate the weather, based on the actions of a fictional person. From the figure it is possible to see that there is a limited number of observations that the model is able to accept: walk, shop, and clean. Based on the observations the HMM receives it is able to give an estimate of what the weather is currently like where the fictional person is. Since the output probability distribution (B) is represented by a matrix, the sum of all elements in a row from this matrix must be equal to 1. Figure 2.1 is created after the HMM example that can be found in [4].

States	=	Rainy, Sunny
Observation	=	Walk, Shop, Clean
Start probability	=	Rainy:0.6, Sunny: 0.4

Transition probability =

	Rainy	*Sunny*
Rainy	0.7	0.3
Sunny	0.4	0.6

Output probability =

	Walk	*Shop*	*Clean*
Rainy	0.1	0.4	0.5
Sunny	0.6	0.3	0.1

Figure 2.1: An example of the components in a discrete HMM. Values taken from Wikipedia [4]

In the case of discrete HMM the input symbols for which the HMM can accept is limited by the input alphabet. However, the input alphabet can be designed to include all kinds of input like letters, numbers, of other symbols. The input alphabet must be defined when the HMM is created. If the input alphabet is changed all elements of the discrete HMM must be updated to accommodate the changes in the input alphabet.

2.1.3 Continuous hidden Markov model

The second version is called a continuous HMM. This version of HMMs is able to handle input that is not part of a predefined input-codebook, but rather input may be any real number. The output probability distribution will be significantly different from a discrete HMM since the input is handled in a different manner.

In a discrete HMM the correct output probability can be found in an output probability matrix, however, in a continuous HMM the output probability must be calculated as the sum from a set of weighted probability distributions, most typically Gaussian distributions, however, other types of probability distributions can be used [15]. Figure 2.2 shows the necessary components of a continuous HMM with three states. In the continuous HMM that Figure 2.2 shows the start probability and transition probability is defined in the same way that it would have been for a discrete HMM. However, when input is received the output probability is calculated as a sum of all the weighted probability distributions, normal distributions in the case of Figure 2.2. For this task, all states will have its own set of weights, mean vectors, and covariance matrices that are used when output probability is calculated.

Number of states = 3

Observations = all real numbers

Start probability =

$State1$	$State2$	$State3$
0.6	0.3	0.1

Transition probability =

	$State1$	$State2$	$State3$
$State1$	0.5	0.2	0.3
$State2$	0.2	0.7	0.1
$State3$	0.1	0.3	0.6

Observation probability = $b_j(O_t)$ = $\sum_{k=1}^{M} c_{jm}$N$(\mu_{jm}, \sum_{jm}, O_t)$

c_{jm} = weighting coefficients
$c_{jm} \geq 0 \quad 1 \leq j \leq N \quad 1 \leq m \leq M$

$\sum_{m=1}^{M} c_{jm} = 1 \quad 1 \leq j \leq N$

μ_{jm} = mean vectors

$\sum_{jm}$ = covariance matrices

O_t = observation at time t

N = total number of states
M = total number of weighted probability distributions

Figure 2.2: The components of a continuous HMM with 3 states.

2.1.4 Three basic HMM inference problems

With HMMs there are basic problems that must be solved if HMM is going to be useful for solving real world problems. These three problems are the following:

- Given an observation sequence O and a HMM λ, how do we find P(O|λ), the probability of the observation sequence, given the HMM?
- Given the observation sequence O, and the HMM λ, how do we find the state sequence Q which best "explains" the observations?
- How do we adjust the model parameters λ = (A,B,π) to maximize the probability of a given observation sequence O (i.e. maximize P(O|λ))?

Solution to problem 1

The answer to the first problem is the forward-backward procedure [12]. From this procedure we can find the forward variable α_t(i) and it is defined as α_t(i) = P(O_1 O_2 $\ldots O_t$, q_t = S_i $|\lambda$). This is the probability of the partial observation sequence, O_1 O_2 $\ldots O_t$,(until time t) and state S_i at time t given the model λ.

The forward-backward procedure also provides us with a backward variable β_t(j) = P(O_{t+1} O_{t+2} $\ldots O_t|q_t$ = S_i, λ) which gives the probability of the partial observation sequence from t+1 to the end, given state S_i at time t and the model λ. Even though the backward variable is not needed for the first problem it becomes useful when solving problem 3[12].

Solution to problem 2

Problem number 2 is a matter of finding the best state sequence that best fits with the observation. The Viterbi algorithm manages this and finds the single best state sequence , Q = (q_1 q_2 $\ldots q_t$), for the given observation O = (O_1 O_2 $\ldots O_t$) [13] [12].

Solution to problem 3

The third problem is to adjust the HMM parameters to maximize the probability of the observation sequence. If given an finite observation sequence there is no optimal way of estimating the models parameters. However, it is possible to chose λ = (A,B,π) such that it is locally maximized for P(O|λ) by using the Baum-Welch algorithm [12]. The Baum-Welch algorithm works by assigning initial probabilities to all the parameters. Then, until the training converges, it adjusts the probabilities of the parameters so as to increase the probability the model assigns to the training set[14].

The Baum-Welch algorithm (or Baum-Welch expectation maximization algorithm) makes use of both the forward variable α_t(i) and the backward variable β_t(j) when it determines updated parameters for the HMM. Because of this the Baum-Welch algorithm is also known as the Forward-Backward algorithm.

To properly estimate the local maximum for P(O|λ) the Baum-Welch algorithm needs several iterations. The algorithm will either be repeated a predetermined number of times, or until the local maximum is found. The local maximum is found when the difference between P(O|λ_{new}) and P(O|λ_{old}) reaches a certain value[12].

2.2 Hidden Markov models in this thesis

While working on this thesis we have made use of several HMMs. To classify error states in a mechanical system several HMMs have been used. To represent the different health states of a mechanical system, one HMM has been created for each individual health state.

All HMMs have been of the type continuous HMM. The type continuous has been used since the observations used for training the HMMs has been represented by numbers. Converting the observation into discrete symbols could lead to loss of information from the observations[12].

The Baum-Welch algorithm has been used to train the HMMs,and the forward variable from the forward-backward algorithm has been used to perform health state detection. The Viterbi algorithm has not been used during this project. If only one HMM had been created for each system the Viterbi algorithm would have been used to perform state detection (where the internal states of the HMM represents the health states of the system). Since one HMM is created for each possible health state the forward variable can be used to determine the health of a system. The HMM (λ) which gives the greatest $P(O|\lambda)$ for a given observation sequence (O) is declared to represent the health the system is currently in.

Chapter 3

Approach

In this chapter, we will present the data used during this project, and how it is possible to perform anomaly detection, system state detection, and make future health predictions of a mechanical component by using hidden Markov models(HMM).

3.1 Data

During this project, we have worked with two different datasets. The first dataset was obtained from Origo Mobikom and contained temperature measurements from an electric motor. The second dataset was generated by us and was created to show how HMMs can be used to predict future states, and to show how multiple states would affect the prediction of future states.

3.1.1 Electric motor temperature data

The first dataset contained temperature measurements taken from four different sensors located in an electric motor.

The data was measured in intervals of 1 minute over a period of several weeks. The temperature measurements had been performed on two identical motors running at the same time in the same system. Both motors had four temperature sensors mounted in them. One of temperature sets was selected to be used as training data, while the other was used to perform anomaly detection and current state detection. Both temperature sets contained 82356 measurements.

Figure 3.1 shows the temperature measurements which will be used to train HMMs, and Figure 3.2 shows the temperature measurements which will be used to test the trained HMMs in anomaly detection and current state detection.

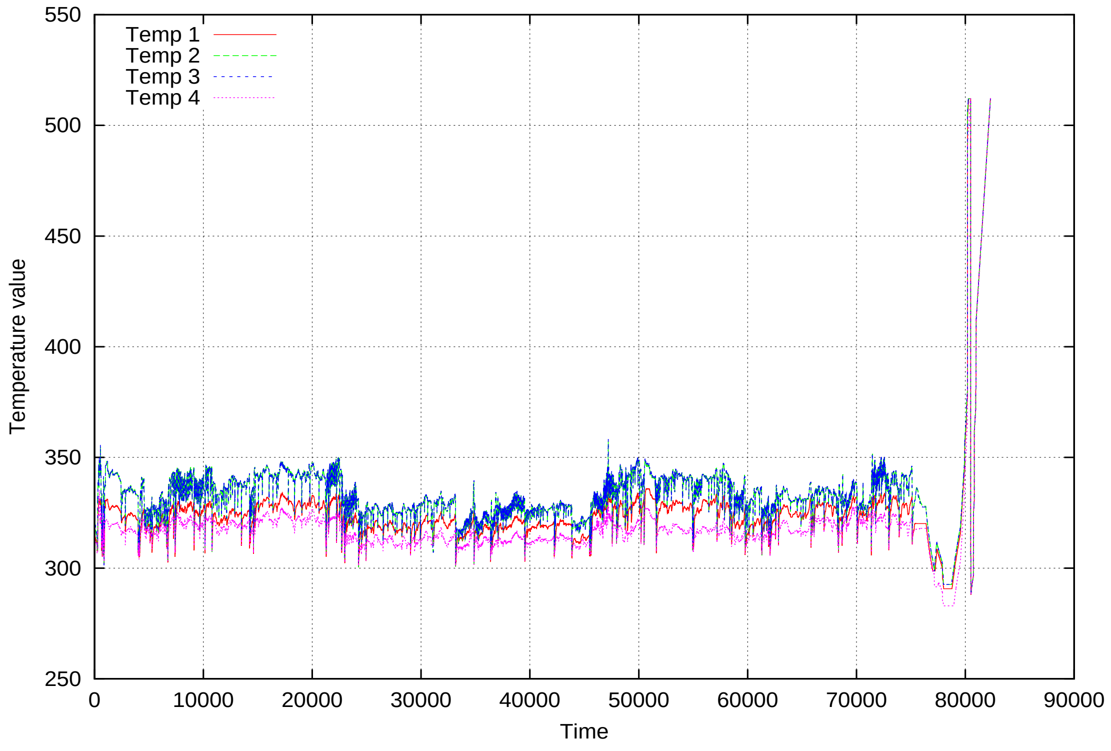

Figure 3.1: Temperature values used for training.

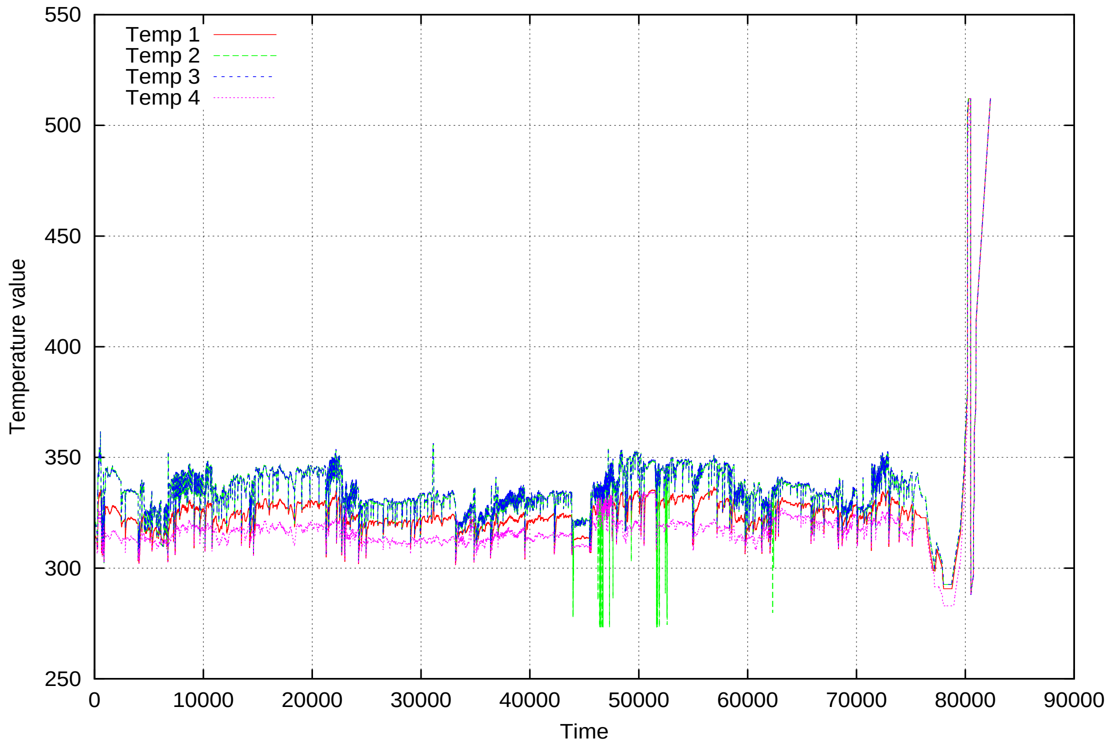

Figure 3.2: Temperature values used for anomaly detection and state recognition.

Because of the lack of expert knowledge it was not possible to say what kind of damage that had occurred. Due to the lack of expert knowledge, and based on the behaviour of the data to-wards the end of the measurements we will assume that values larger than 400 indicates that something is wrong, and would mean that the system is in need of maintenance. This means that the created HMMs will represent two states, good-state and bad-state.

Unfortunately it is impossible to say whether the sudden temperature rise was caused by a component malfunction or by some outside event. The difference here is that component malfunction may be monitored and predicted. However, if something gets caught in the intake valve for the pump which the electric motor is running(or some other event) it would be impossible to predict. Since the data was gathered at the same time in two different motors running in the same system, we will assume that it was some kind of outside influence that caused the temperature to rise. Based on this we decided to only use this data for anomaly detection and current state detection.

3.1.2 Generated data

The generated dataset consists of four separate observation vectors. The three first observation vectors consists of 3000 observations taken at random from a normal distribution with added noise. The three first observation vectors of the generated data is show in Figure 3.3. Out of these three generated vectors, vector 1 and vector 3 will be used as training data, vector 2 will be used to perform future state predictions. The generated data can not be said to represent data from a specific mechanical component, or represent the change in measured data that some components may have. However, it can be divided into several states which is one of the most important features for data where future states is going to be predicted.

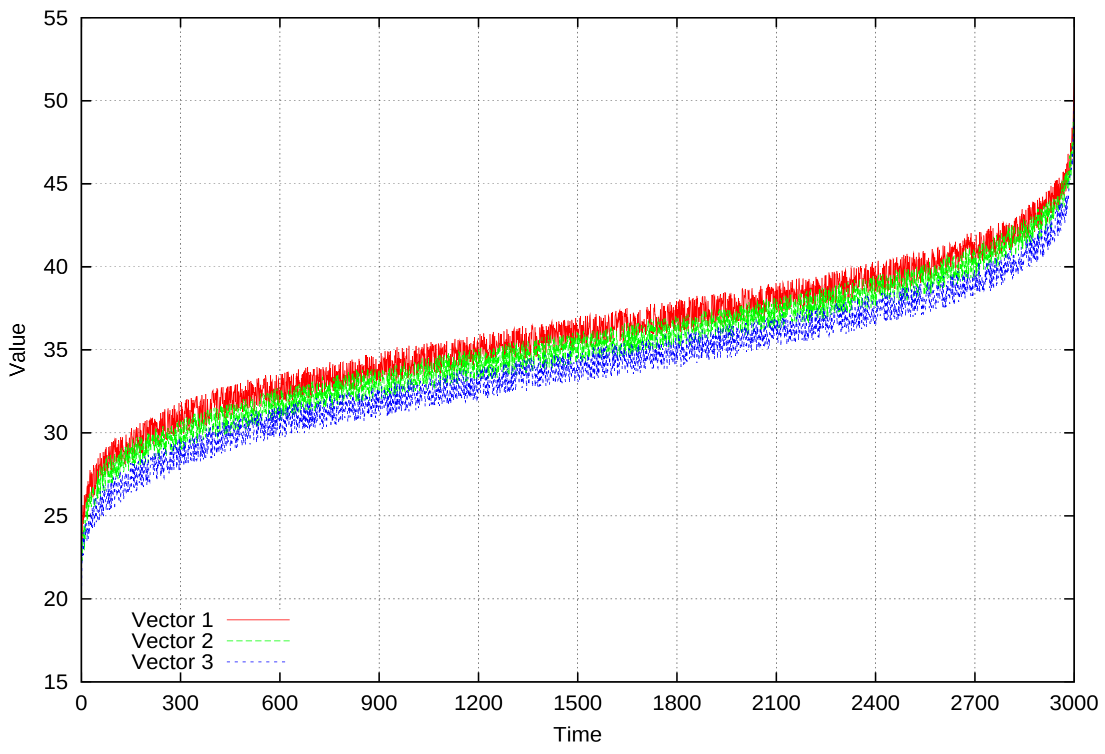

Figure 3.3: The generated observation vectors.

In addition to the generated data shown in Figure 3.3 a fourth observation vector was also generated. This vector is shown in Figure 3.4. It was designed to be much longer so that the state transition points would be vastly different from any of the training data that is used to estimate the initial prediction parameters for future state prediction. In this way it should be possible to show how prediction is affected when it tries to predict transition points which has not been seen during training.

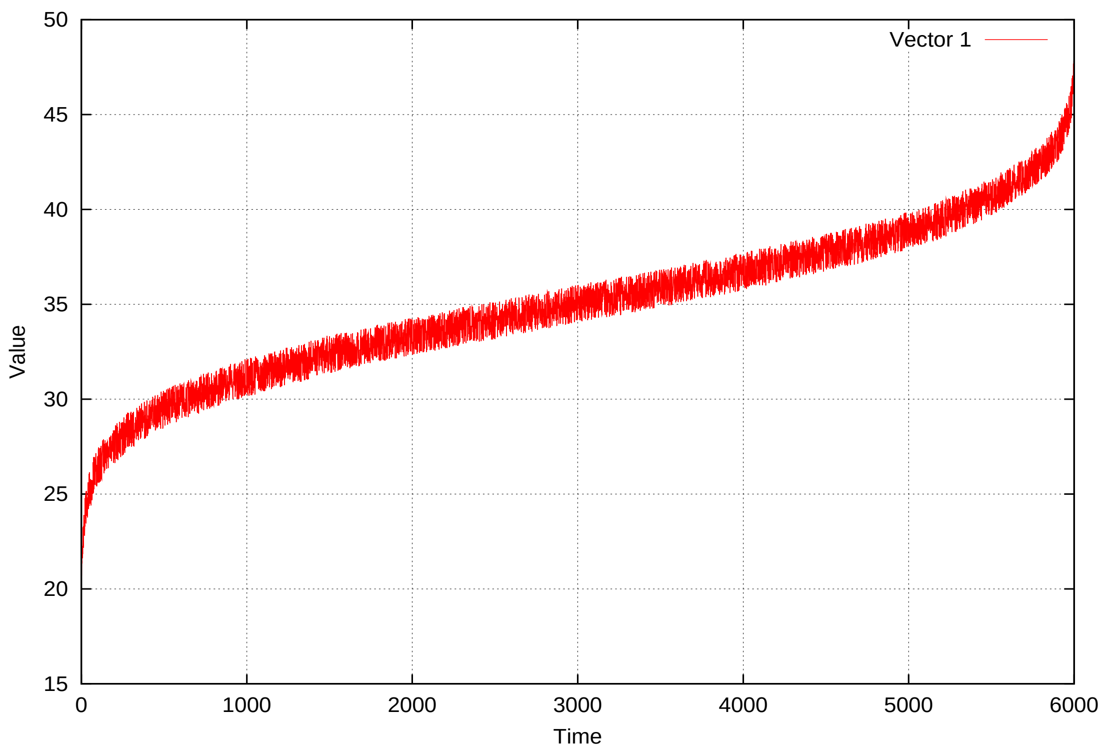

Figure 3.4: The fourth generated observations vector.

Even thought the generated data does not necessarily represents a real life system, it still represents the kind of data which a HMM based prediction will be best suited for. The kind of system most suited for HMM predictions is one which goes through several different health states. The more health-states the system goes through the more accurate the prediction has potential to be. As shown later in this chapter the generated data will be divided into 4 different health states.

3.2 Hidden Markov Models

When creating HMMs during this project we used a HMM implementation for Matlab called “Hidden Markov Model (HMM) Toolbox for Matlab”[9].

All HMMs created used during this project were of the same type, with the same number of states, and with the same number of weighted Gaussian distributions. The type of HMM had to be continuous since the observation data consisted of a series of numbers, and loss of information might have occurred if the data was discretized to make them usable by a discrete HMM[12].

One of the first choices was whether to represent the system as one HMM or several HMMs. If one HMM is used, each state in the HMM will represent a different health state for the mechanical system. If several HMMs are used, each

HMM will be trained to represent a different health state for the system. Using only one HMM might be optimal when a lot of data is present at the time of creation. However, using several HMMs to represent system health states might be better if little data is available. When more data becomes available and new health states are identified, a new HMM can be trained and added to the system of existing HMMs.

For this thesis the decision was made to represent the health states of each system as a collection of several HMMs, even though the same results could have been achieved by using only a single HMM for each system. The test for current state detection, will be slightly different based on the multi HMM approach. The difference in approach will be noted in the section for current state detection.

The first step in creating the HMMs was to determine how many health states they where going to represent. To represent the possible health states of the electric motor two HMMs have been trained. One HMM will represents the motor when it is in good health, and the other HMM will represents the motor while it is in bad health. These two HMMs will be referred to as $HMM - 1_{good}$ and $HMM - 1_{bad}$. If more detailed information had been present about what the data actually represents it might have been possible to create more HMMs so that the health-states of the system could be more accurately represented.

Figure 3.5 shows how the electric motor temperature data was divided between the HMMs during training. One square shows the data used when training $HMM - 1_{good}$, while the other square shows the data used for training $HMM - 1_{bad}$.

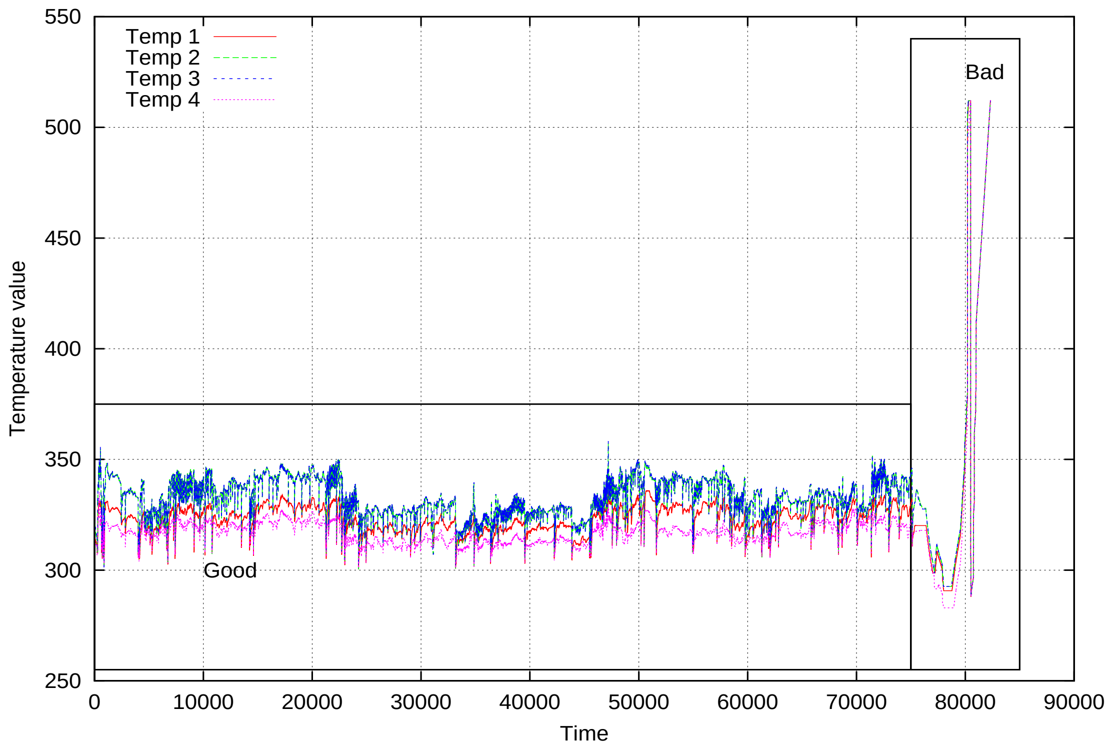

Figure 3.5: Shows how the data was divided when HMMs were being created for the temperature data.

To determine the number of states and weighted gaussians a test was performed to determine the number which was best with regards to learning accuracy. The test was performed for 2 - 6 states and for 2 - 5 weighted gaussians. After training each HMM was presented with a series of observation sequences taken from their own training data to determine how well it would recognize the data. Normally when training HMMs, testing of learning accuracy is performed on observation data which was not used to train the system. However, due to the small amount of available data the recognition accuracy was performed on the same data that was used for training. This should be avoided when possible since the HMMs will be better suited to recognize their own training data, while it is most interesting to see how well new data is recognized.

Table 3.1: Training accuracy for different combinations of number of states and gaussians.

States	Gaussians	accuracy good	accuracy bad	training time (sec)
2.000000	2.000000	-426.441766	-313.040668	318
2.000000	3.000000	-411.716676	-372.475489	394
2.000000	4.000000	-408.144722	-251.444811	451
2.000000	5.000000	-413.857508	-265.292718	581
3.000000	2.000000	-391.499151	-377.190206	405
3.000000	3.000000	-402.105029	-393.349187	403
3.000000	4.000000	-386.764117	-286.442257	441
3.000000	5.000000	-385.067205	-158.739353	550
4.000000	2.000000	-376.154570	-451.001756	508
4.000000	3.000000	-372.914253	-160.767891	701
4.000000	4.000000	-377.764539	-85.606621	895
4.000000	5.000000	-371.759200	-163.589055	836
5.000000	2.000000	-373.973499	-399.154423	459
5.000000	3.000000	-364.722211	-215.544102	1090
5.000000	4.000000	-359.473372	-191.456421	835
5.000000	5.000000	-352.054112	-200.105980	609
6.000000	2.000000	-360.779454	-264.582630	721
6.000000	3.000000	-360.448389	-273.858578	904
6.000000	4.000000	-343.358507	-150.069212	738
6.000000	5.000000	-346.892211	-121.916155	1120

Table 3.1 shows the result of this test and it is possible to see that a higher number of states and gaussians provide a better recognition (a value closer to 0 indicate better recognition). The recognition accuracy for $HMM-1_{good}$ remains reasonably consistent for each state/Gaussian combination and shows only small improvements in recognition accuracy. However, $HMM-1_{bad}$ had a huge increase in recognition accuracy when it was trained for states $\geq$ 4 and gaussians $\geq$ 3. Closer inspection revealed that $HMM-1_{bad}$ would produce positive log-likelihood for some of its training data. However, all log-likelihood values should be less than 0 since any value greater than 0 would mean that the recognition accuracy was over 100%. The cause of positive log-likelihood may be due to the variation of values in the training data for $HMM-1_{bad}$[10] which ranges from 290 to 512 and prevents $HMM-1_{bad}$ from being trained properly. $HMM-1_{good}$ produced negative log-likelihood values for all combinations of states and gaussians, and it is possible that the number of states and Gaussians may be increased even further to produce even better recognition. However, to keep the models equal we decided to create the HMMs with 4 states and 2 weighted gaussians.

The total training time used to train both HMMs have also been included in Table 3.1. This is to show that training time increases when the number of states and/or weighted Gaussians increase. However, training time should not be an important factor when choosing the number of states/weighted-Gaussians since the HMMs only need to be trained once.

When deciding on the number of states is it also important to consider that over-fitting might occur. Over-fitting will happen when a HMM is given more states than what is necessary to properly learn the data. An over-fitted HMM might adjust to very specific random features of the training data[2].

A similar approach was taken when training HMMs on the generated data. However, for this dataset four different HMMs were created to represent four different health states. Each HMM had 4 states and 3 weighted gaussians. Values less than 30 were decided to represent the good state, values from 30 to 35 were decided to represent the medium-good state, values from 35 to 40 represents the medium-bad state, while values larger than 40 represents the bad state. The training vectors as well as the different training areas are shown in Figure 3.6. The four different HMMs trained for this dataset will be referred to as $HMM-2_{good}$, $HMM-2_{medium-good}$, $HMM-2_{medium-bad}$, and $HMM-2_{bad}$.

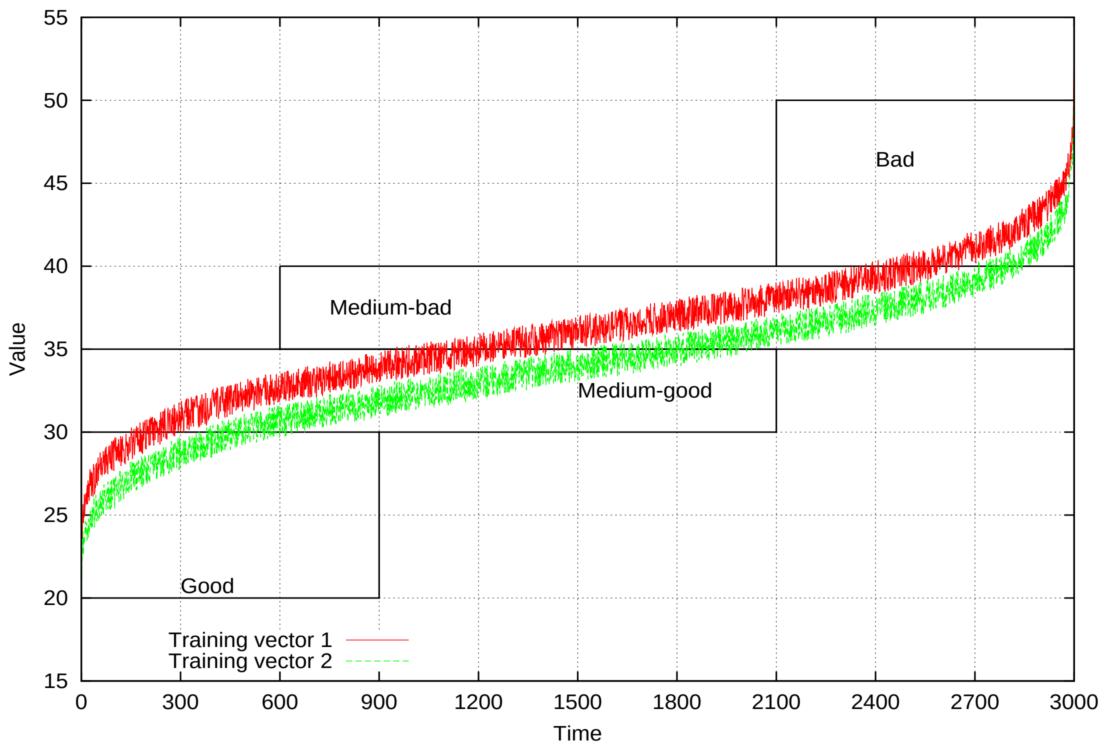

Figure 3.6: Shows how the data was divided when HMMs were being created for the generated data.

Only two of the four generated data vectors was used to train the HMMs for the generated data. Two training vectors, where each training vector represents a complete component life cycle, is the minimum amount of training data required to be able to perform future state prediction. If only one life cycle of observations was present, the only prediction that could be made about future state transition points is the same state transition points as the one in the available training data. A minimum amount of data (two vectors)) has been used simply because it is seldom possible to obtain a lot of data for creating CBM systems, and training HMMs.

3.3 Anomaly detection, current state detection, and future state prediction

When the HMMs had been trained it is possible to use them to perform anomaly detection, detect which state the system is currently in, and estimate the remaining useful life of the system by predicting future state transitions.

When performing anomaly detection and current state detection only the electric motor temperature data will be used. The reason for this is that the temperature data represents data from a real mechanical system and the method for performing anomaly detection and state recognition will be the same regardless of which dataset is used.

For future state prediction the generated data will be used to show how prediction can be performed. The reason for this is that future state prediction is best to perform on measurements that show a more gradual approach toward system failure, preferably through several different health states.

3.3.1 Anomaly detection

When analyzing data from a mechanical system, anomaly detection refers to discovering data which no HMM has been trained to recognise. When a HMM is presented with an observation vector, it is possible to determine the probability that the observation vector “belongs” to the HMM by using the forward-algorithm.

To show this $HMM-1_{good}$ will be used. This will represent that the only training data available is the data that show when the system is in good health. $HMM-1_{good}$ will be presented with observation vectors containing 60 observations. The first observation vector will contain measurements 1 - 60, the next observation vector will contain measurements 2 - 61, and so on. The observations are taken from the temperature measurements that were not used to train the HMM.

Figure 3.7 shows the log-likelihood that the received status data belongs to $HMM-1_{good}$. In the beginning only good status data is received, this is shown in the figure since the likelihood is closer to 0. In this case a value of 0 would mean that it was a 100% chance that the received data belonged to $HMM-1_{good}$. However, when $HMM-1_{good}$ starts receiving data that would indicate that something is wrong the likelihood drops. Once the likelihood drops below a predefined value it is possible to say that the system is not in good health any longer. However, once the likelihood is below this predefined value it is not possible to say what kind of state the system is in, it is only possible to say that it is no longer in the good-health-state.

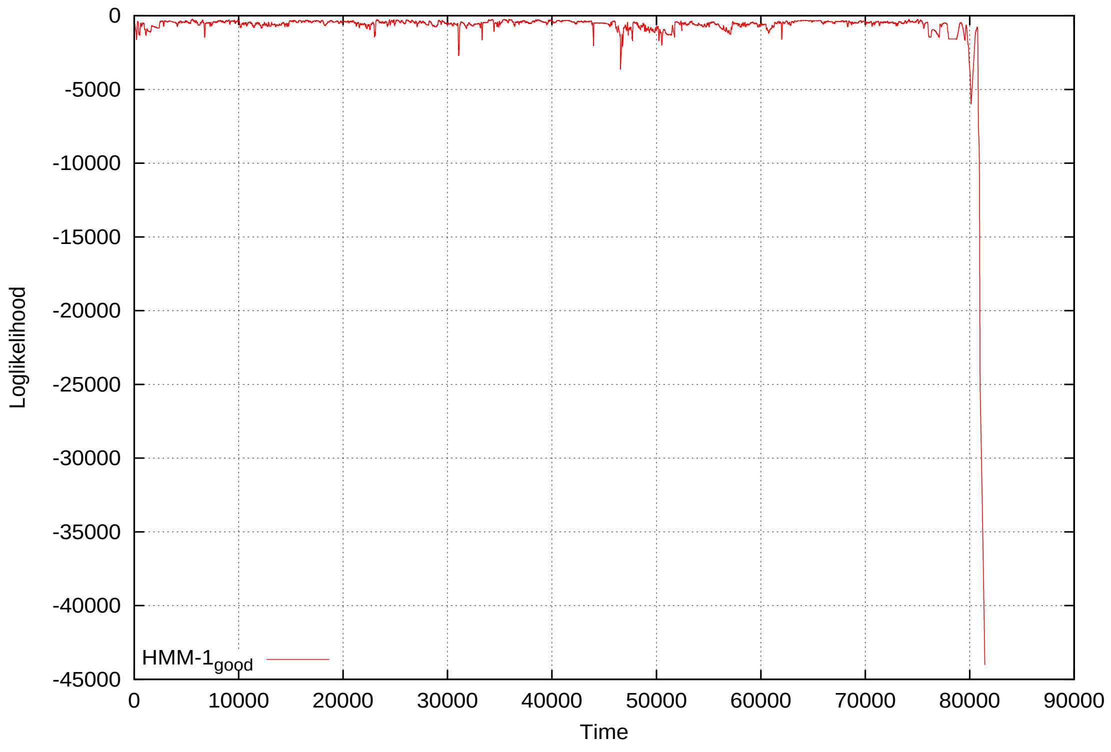

Figure 3.7: The log-likelihood that the received status data belongs to $HMM - 1_{good}$.

Anomaly detection is performed in the same way if a single HMM has been trained to recognize data. The HMM is presented with a series of observation vectors and once the HMM receives data which it has not been trained to recognize, the log-likelihood will drop in the same way as in Figure 3.7.

3.3.2 State detection

It is possible to determine the state which the system is currently in once several HMMs have been trained. To show how state detection is performed $HMM-1_{good}$ and $HMM - 1_{bad}$ will be used.

In this case both HMMs will be presented with observation vectors containing 60 observations. The first observation vector will contain measurements 1 - 60, the next observation vector will contain measurements 2 - 61, and so on. The observations are taken from the data which was not used to train the HMMs.

Since both HMMs are presented with the same observation vector it can be said that they are competing against each other. The log-likelihood will be used to determine which of the two HMMs is the winner. The HMM which produces the best log-likelihood (i.e. closest to 0) is declared to be the winner, and the system is declared to be in the state which the winning HMM represents. However, even after the winner is declared it is not 100% certain that the system is in the state which the winning HMM represents. The winner of the log-likelihood competition only determines the most likely state which the system is in.

Figure 3.8 shows how the likelihood for each state changes over time. After a certain time $HMM-1_{bad}$ will produce a likelihood that is better than $HMM-1_{good}$. Once $HMM-1_{bad}$ produces the best likelihood, it is declared that the system has gone from state good to state bad. The state transition point is the point in Figure 3.8 where both likelihood trajectories intersect each other.

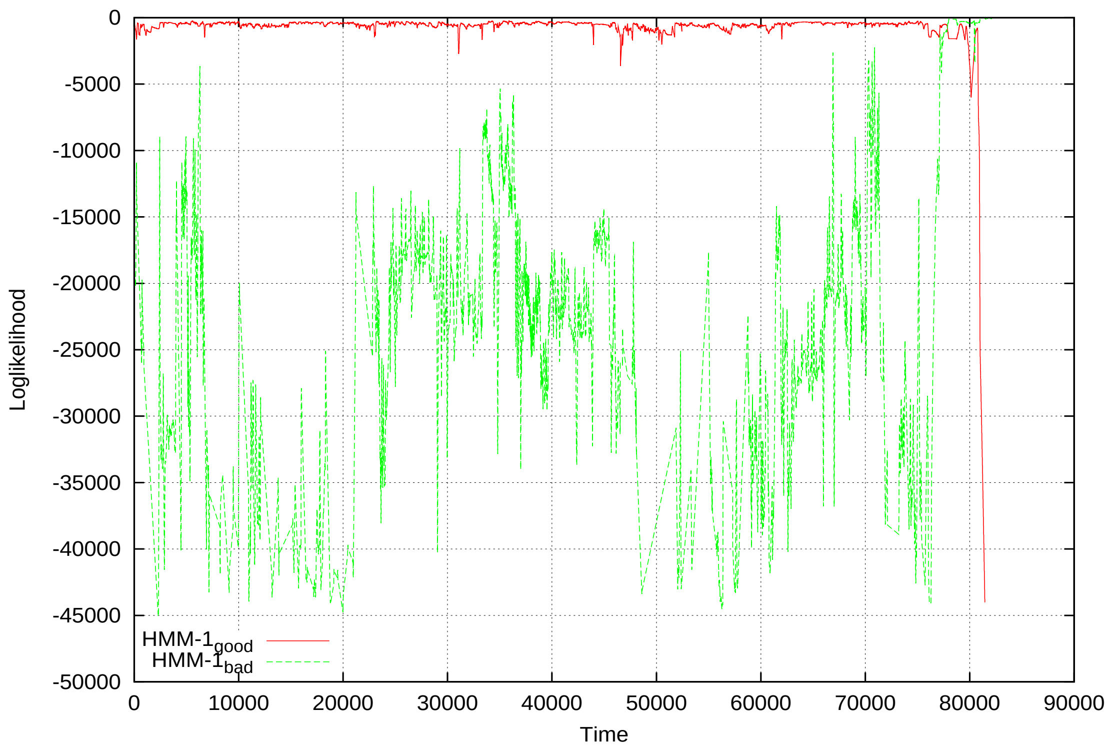

Figure 3.8: The likelihood is used to determine when the system goes from good to bad. Observations are taken from the electric motor data that was not used to train the HMMs.

It is possible for the log-likelihood trajectories to intersect multiple times (due to noisy measurements or other factors), especially in the area where the state transition takes place. For this test we have decided to record the first point where the likelihood trajectories intersect, and from this point onward the system will be declared to be in the bad-state.

If current state detection is being performed on a system were all health states is being represented by a single HMM, then it is not possible to determine the current state by holding a best log-likelihood competition. However, the Viterbi algorithm is designed to take a series of observations, and determine the state sequence that best fits with the given observations. By using the Viterbi algorithm on the available observations, one will obtain the single best state sequence that best describes the observations. In other words, the Viterbi algorithm will give a sequential list of the health states that the observations represents.

For a single HMM trained on the same data as $HMM-1_{good}$ and $HMM-2_{bad}$ the state sequence received from the Viterbi algorithm might not just consist of two states. Since the single HMM will require several states and weighted gaussians to be able to learn the system properly. Because of the possible requirement of several states in the HMM then each health state is probably represented by several states. This is something that has to be handled during creation of the HMM.

When using several HMMs where each HMM represents a single health state, the number of states in each HMM is not interesting. The only thing that is interesting is how well the HMMs can recognize the data that corresponds with the health state that it is supposed to represent.

3.3.3 Future state prediction

The aim of future state prediction is to estimate when the system is going to change from one state to another. It is impossible to predict the exact moment when a system is going to change states. However, it should be possible to give a reasonable estimation about when the system can be expected to go from one state to another[5]. It is also possible to estimate a probability that indicates the chance that the system is going to change states in a certain amount of time.

Future state prediction is done by estimating the state transition point to the final state. In this case we will assume that the final state transition is the one of interest since the component will be in need of repair once the final state transition point is detected. However, if the last state transition point is not the one of interest the proposed method can be used to estimate all future state transitions. A weakness of the proposed state prediction method is that the first state transition needs to be recorded before a prediction can be made. This is because the prediction method needs some information from the system to determine the final state transition point.

The future state prediction presented here is the same that is used in[5]. The proposed prediction solution in[5] uses a method that can predict the final state transition point, while using all previous transition points to adjust the prediction. However, in this case a lot less training data is used, and we will show how this will effect the prediction.

To estimate state transition points it is first necessary to find the state transition points from the training data. These points are determined in the same way that current state prediction is made. Figure 3.9 and Figure 3.10 show the log-likelihood trajectories for the generated data that was used to train $HMM-2_{good}$, $HMM-2_{medium-good}$, $HMM-2_{medium-bad}$, and $HMM-2_{bad}$. In these figures the state transition points are marked with a black line. By using these transition points it is possible to give an estimated point indicating where state transition is likely to happen. The state transition points are summarized in Table 3.2.

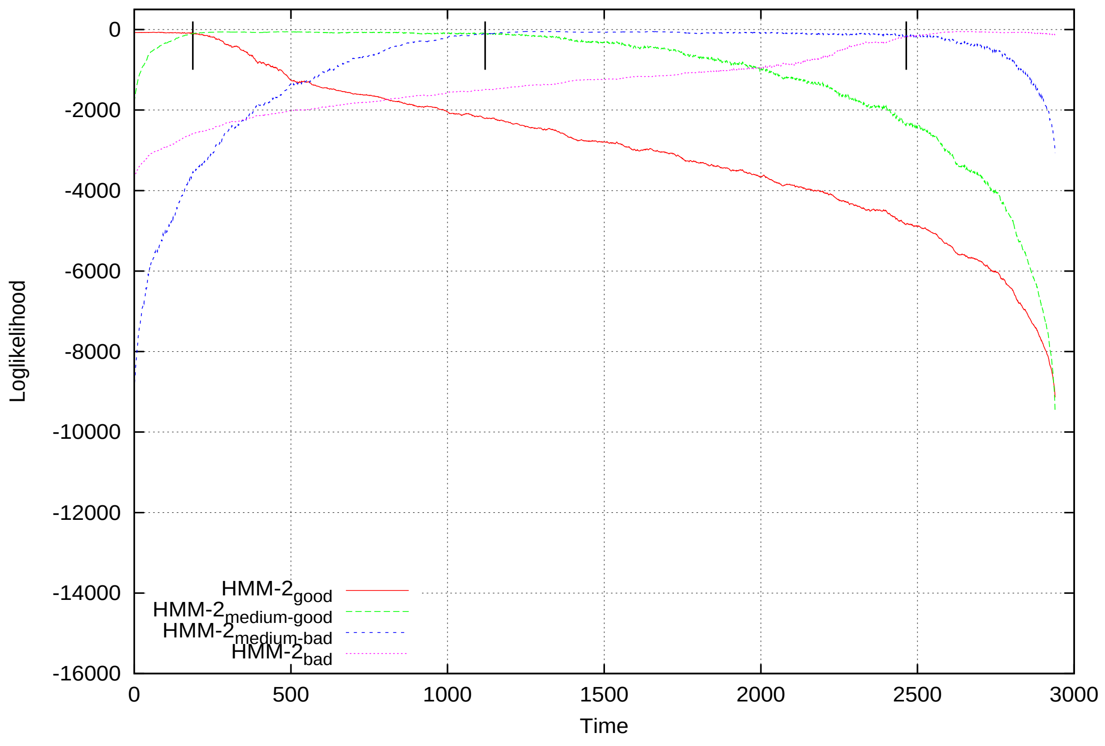

Figure 3.9: State transition points for the first training vector.

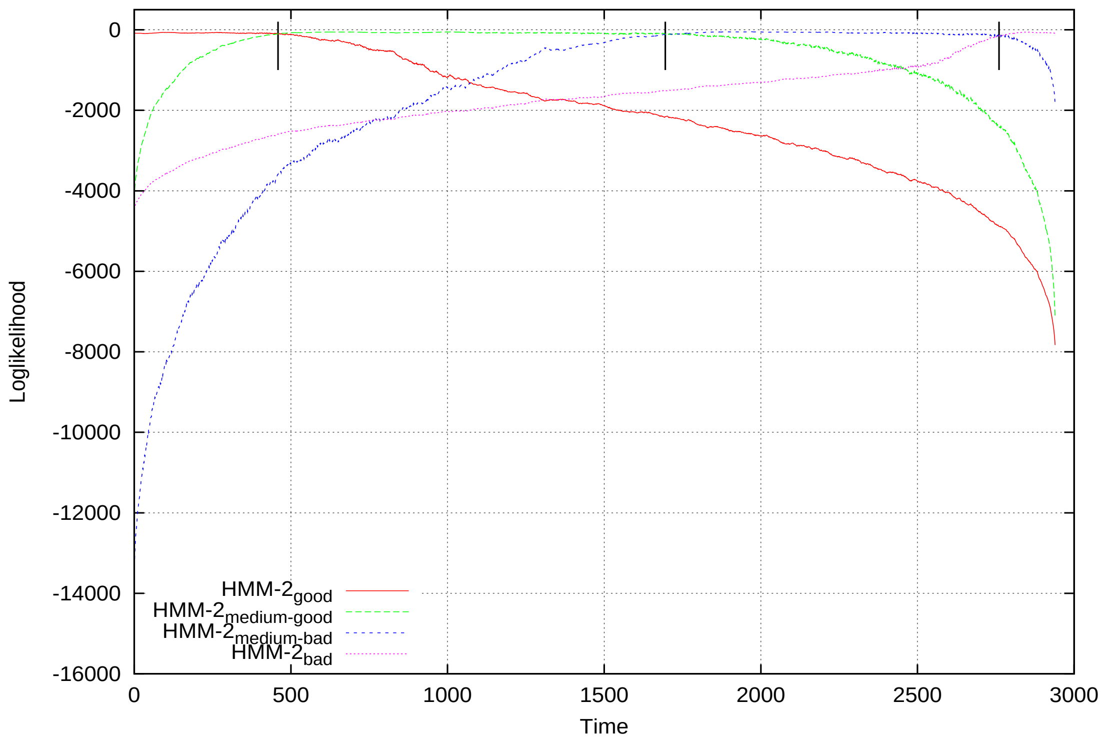

Figure 3.10: State transition points for the second training vector.

Table 3.2: The state transition times for all states from good to bad for the different training vectors.

Training vector	good $\rightarrow$ medium-good	medium-good $\rightarrow$ medium-bad	medium-bad $\rightarrow$ bad
1	187	1120	2464
2	459	1695	2760

The basic theory about future state prediction is as follows: Lets assume that observation sequences are available from R similar units for the purpose of developing state detection and state prediction models. This will result in R estimated vectors of state transition points, T = [S_1, S_2,. . . ,S_R]. The fundamental assumption is that T contains the necessary information to provide future state prediction. The procedure is as follows. Start with the assumption that T follows some multivariate distribution. Once the distribution is assessed, the conditional probability distribution for a specific state transition $t_{S_i \rightarrow S_{i+1}}$ given the previous state transition points ($S_1 \rightarrow S_2$,. . . ,$S_{i-1} \rightarrow S_i$) can be estimated. The process can be iterated in a recursive manner to make predictions regarding several sequential state transitions[5].

Once the state transition points of the training data has been established it is possible to model the joint distributions and conditional distributions that can be used to perform future state prediction. Since there are four different states, a total of 3 state transition points are recorded for each training vector, namely $[t_{good \to medium-good}, t_{medium-good \to medium-bad}, t_{medium-bad \to bad}]$. With small datasets it is best to treat them as normally distributed since the probability distribution can not be determined easily, for large datasets the probability distribution may be properly determined by extensive investigation. For determining the multivariate distribution of our datasets state transition points, we will for the purpose of this master thesis assume that they follow a Gaussian distribution, even though they could potentially follow any kind of probability distribution.

After the distribution of the state transition points are determined, it is possible to determine the joint distributions which are necessary for the estimation of parameters for the conditional distributions that can be used for future state prediction. The joint distributions which are necessary are the following:

- $[t_{good \to medium-good}, t_{medium-bad \to bad}]$. This joint distribution allows the final state transition point to be estimated once the first state transition point is detected. As such this joint distribution allows us to estimate the parameters for the following conditional distribution $f(t_{medium-bad \to bad} \mid t_{good \to medium-good})$.
- $[t_{good \to medium-good}, t_{medium-good \to medium-bad}, t_{medium-bad \to bad}]$. From this joint probability distribution it is possible to determine the parameters of the conditional probability $f(t_{medium-bad \to bad} \mid t_{good \to medium-good}, t_{medium-good \to medium-bad})$ once $t_{good \to medium-good}$ and $t_{medium-good \to medium-bad}$ become known.

According to [5] one of the basic properties of the Gaussian distribution is as follows: If X = $[X_1{:}X2_2]$ is distributed as N(μ,Σ) with μ = $[\mu_1{:}\mu_2]$, Σ = $\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$ and $\mid \Sigma_{22} \mid \geq$ 0, then the conditional distribution of X_2 given X_1 = x_1 is also a Gaussian with a mean vector of μ_2 + $\Sigma_{21} * \Sigma_{11}^{-1} * (x_1 - \mu_1)$ and a covariance matrix of $\Sigma_{22} + \Sigma_{21} * \Sigma_{11}^{-1} * \Sigma_{12}$.

The above property can be used to find all conditional distributions of interest. For this thesis the most interesting conditional distributions are $f(t_{medium-bad \to bad} \mid t_{good \to medium-good})$ and $f(t_{medium-bad \to bad} \mid t_{good \to medium-good}, t_{medium-good \to medium-bad})$, however, all state transitions can be determined in this way. Only two conditional distributions will be considered since they will attempt to predict the last state transition. The last state transition is of most interest since it can be used to determine when the system need maintenance (i.e bad-state = useless component)[5].

Figure 3.11 shows the necessary parameters which must be determined before the conditional distributions can be estimated. For details on how to determine the parameters in Figure 3.11 from the state transition points estimated from training data see[8]. In Figure 3.11 X_1, X_2 and X_3 are the gathered state transition points from the available training data, while m1-m3 and std1 - std3 contain the mean values and standard deviation for X_1 - X_3. U1 - u3 are the mean vectors containing the expected values for X_1 - X_3. The covariance matrix in Figure 3.11 contain all covariance matrices for the state transition points X_1 - X_3. Together all the values in Figure 3.11 contain all the necessary information to estimate all the necessary conditional distributions that are necessary for predicting future state transition points. X_1 - X_3 contain all the information needed to estimate all parameters in Figure 3.11.

X_1 = [187,459] = state transition points for $t_{good \rightarrow medium-good}$
X_2 = [1120,1695] = state transition points for $t_{medium-good \rightarrow medium-bad}$
X_3 = [2464,2760] = state transition points for $t_{medium-bad \rightarrow bad}$
m1 = 323, m2 = 1407.5, m3 = 2612
std1 = 136, std2 = 287,5, std3 = 148
u1 = [29.6685 , 386.1772]
u2 = [809.9 , 1426.6]
u3 = [390.9 , 2322.1]

Covariance matrix for X_1, X_2, and X_3 = $\begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{bmatrix}$ =

$$\begin{bmatrix} 3930 & 1530 & 5600 & 5640 & 8210 & 9200 \\ 1530 & 04460 & 13740 & 13840 & 20150 & 22570 \\ 5600 & 13740 & 69530 & 50650 & 73750 & 82610 \\ 5640 & 13840 & 50650 & 60630 & 74300 & 83220 \\ 8210 & 20150 & 73750 & 74300 & 681840 & 121170 \\ 9200 & 22570 & 82610 & 83220 & 121170 & 161320 \end{bmatrix}$$

Each Σ_{xy} in the Covariance matrix is in this case a 2x2 matrix, and each u is a 2x1 vector. The sizes of these would grow if more state transition points were available.

Figure 3.11: Necessary parameters for estimating the parameters for conditional distributions which can be used for future state prediction.

Based on the information shown in Figure 3.11 the necessary conditional distributions are created. The conditional distribution $f(t_{medium-bad \rightarrow bad} \mid t_{good \rightarrow medium-good})$ will have a mean vector ofu_3 + $\Sigma_{31} * \Sigma_{11}^{-1} * (T1 - u_1)$ and a covariance matrix of $\Sigma_{33} + \Sigma_{31} * \Sigma_{11}^{-1} * \Sigma_{13}$ were $T1$ is the first state transition point from a new vector of observations. The second conditional distribution, $f(t_{medium-bad \rightarrow bad} \mid t_{good \rightarrow medium-good}, t_{medium-good \rightarrow medium-bad})$, will have a mean vector u_3 + $\Sigma_{32} * \Sigma_{X1|X2}^{-1} * (T2 - M2)$ and a covariance matrix $\Sigma_{33} + \Sigma_{32} * \Sigma_{X1|X2}^{-1} * \Sigma_{23}$ were $M2$ and $\Sigma_{X1|X2}$ is the mean vector and covariance matrix for the conditional distribution $f(t_{medium-good \rightarrow medium-bad} \mid t_{good \rightarrow medium-good})$ and $T2$ is the second state transition point from a new vector of observations. The mean vector and covariance matrices for both conditional distributions is shown in Figure 3.12.

$f(t_{medium-bad \to bad} \mid t_{good \to medium-good})$

estimated mean vector = [390.9 , 2322.1] + $\begin{bmatrix} 0.3828 & 4.3853 \\ 0.4287 & 4.9121 \end{bmatrix}$ * (T1 - [29.6685 , 386.1772])

estimated covariance matrix = $\begin{bmatrix} 773360 & 223680 \\ 223680 & 276150 \end{bmatrix}$

$f(t_{medium-bad \to bad} \mid t_{good \to medium-good}, t_{medium-good \to medium-bad})$

estimated mean vector = [809.9 , 1426.6] + $\begin{bmatrix} 0.24510.4950 \\ 0.27450.5545 \end{bmatrix}$ * (T2 - ([390.9 , 2322.1] + $\Sigma_{31} * \Sigma_{11}^{-1}$*(T1 - [29.6685 , 386.1772]))

estimated covariance matrix = $\begin{bmatrix} 112060 & 93500 \\ 93500 & 103800 \end{bmatrix}$

Figure 3.12: Estimated parameters for the conditional distributions that will be used to predict the final state transition.

It should be possible to see that the parameters in Figure 3.12 takes into account the transition points that are observed from new observations when making prediction, and that $f(t_{medium-bad \to bad} \mid t_{good \to medium-good}, t_{medium-good \to medium-bad})$ takes into account both T1 and T2 to adjust the prediction. It can also be seen that only the mean vectors of the different conditional distributions are dependent on new observations. The covariance matrices can be calculated entirely from the training data.

When performing future state transition point prediction more states will lead to a more accurate prediction, since more information will be used from the component under investigation[5]. The prediction will give a result as X-number of observations until the final state change. Because of this it is necessary to know how often observations are made (i.e. every second, minute or hour).

Chapter 4

Results and analysis

In this chapter the results of the tests that were performed with regards to anomaly detection, current state detection, and future state prediction will be presented.

4.1 Anomaly detection

The first test was using a trained hidden Markov model (HMM) to perform anomaly detection. $HMM - 1_{good}$ was presented with observations and the log-likelihood was recorded as new observations were presented. Figure 4.1 shows the recorded log-likelihoods as well as the areas where it can be said that anomalies were detected. The first two anomalies was consistent with drops in temperature from the measured data. Someone with expert knowledge about the system might have been able to say if the two first anomalies was something that should have been investigated more closely as possible errors or indicators about future errors. However, without such knowledge we will assume that these anomalies are part of normal operating conditions.

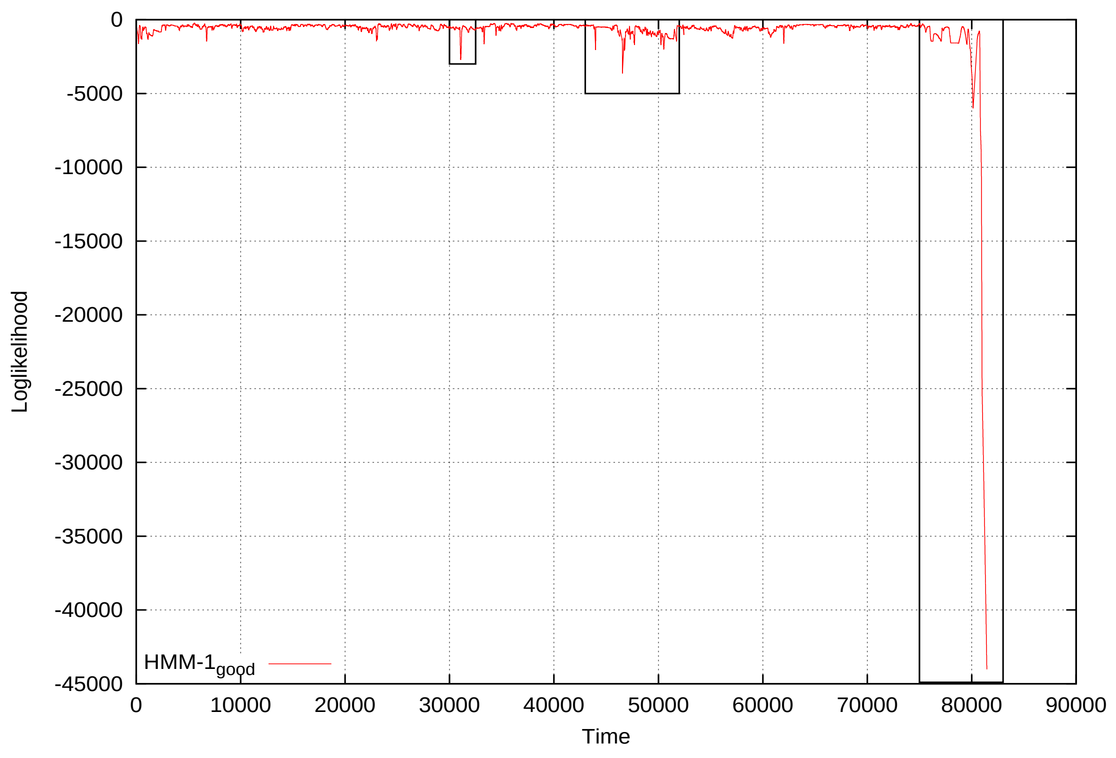

Figure 4.1: The results of anomaly detection performed on $HMM - 1_{good}$.

The third (and last) anomaly in Figure 4.1 comes from the data that was designated as bad. The log-likelihood goes through a short and slow decent and then plummets down. Based on the result presented in Figure 4.1 we will draw the conclusion that using HMMs as a way of performing anomaly detection is a viable option for CBM systems. Even when there is limited training data it should still be possible to perform anomaly detection in a satisfactory manner. However, without expert knowledge several anomalies might might be classified as part of normal working conditions, rather than possible errors or future error indicators.

For training a HMM to perform anomaly detection little training data is required. However, there should still be some knowledge about what kind of health-state(s) the delivered data represents. Since there are limited amounts of training data available for anomaly detection systems there should be better access to expert knowledge to better determine the cause of the anomalies that are detected.

4.2 Current state detection

When detecting the current state of the electric motor both $HMM - 1_{good}$ and $HMM - 1_{bad}$ was used. This test attempted to determine whether the two HMMs manages to accurately determine the current state of the system. Both HMMs were presented with the same observations and based on the log-likelihood that each HMM produced it should be possible to determine the current state of the system. In this case the current state also determines the current health, were $HMM - 1_{good}$ indicates good health and $HMM - 1_{bad}$ indicates bad health.

The log-likelihood trajectories in Figure 4.2 shows how the log-likelihood of each HMM changes as new observations comes in. It is evident from this figure that the log-likelihood of $HMM - 1_{good}$ is much more stable than the log likelihood of $HMM - 1_{bad}$. The reason for this is that the training data for $HMM - 1_{bad}$ includes several measurements in the range of 300-350 which is exactly the same as most of the training data for $HMM - 1_{good}$.

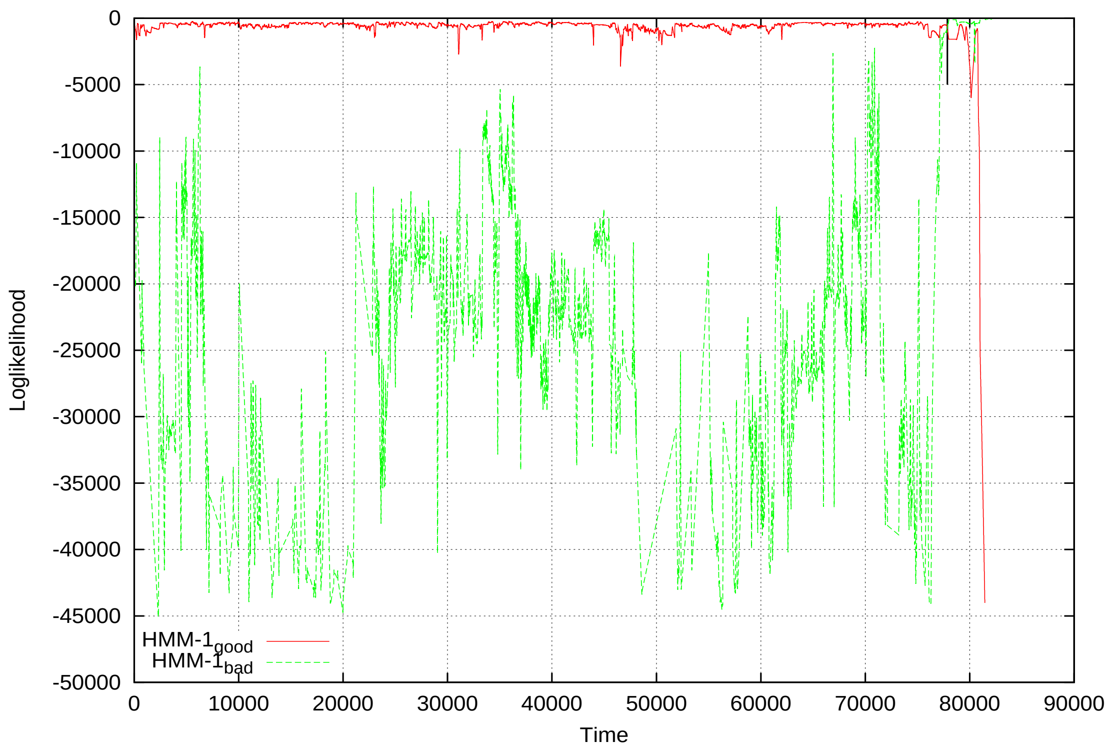

Figure 4.2: The log-likelihood is used to determine when the system goes from good to bad. Observations were taken from the vector that was not used to train the HMMs.

Based on the log-likelihood intersection in Figure 4.2 the electric motor went from state-good to state-bad at observation 77866. However, when taking a closer look on the actual observations, the actual state transition should have happened around the 77029th observation. This means that the state transition was delayed by about 800 observations. Since the observations were taken at 1 minute intervals this means that the state transition was reported about 13 hours late. This is most likely caused by the division of training data. If $HMM - 1_{bad}$ had been given more concentrated training data the result would probably have been better.

Based on the results presented in Figure 4.2 we can say that HMMs are definitely suited to perform current state prediction, however, the training data will have to be divided better between the HMMs which will perform state detection. The amount of training data must then also be sufficient to allow all HMMs to be properly trained to recognize data belonging to them.

A test was conducted to see if 3 HMMs could be trained on the same training data to see if this would improve the current state detection. $HMM-1_{good}$ was trained on the same data as before. However, $HMM-1_{bad}$ received a smaller amount of training data than before, while a third HMM ($HMM-1_{medium}$) was trained on the remaining data. $HMM-1_{bad}$ was restricted to only receiving training data with values larger than 400, while $HMM-1_{medium}$ was only given training data with values smaller than 300. Prior to this test there was not performed any test to determine the learning accuracy based on combinations of states and Gaussians. Instead all 3 HMMs were created with 4 states and 2 weighted Gaussians, exactly as when the first state detection test was performed.

The log-likelihood for the 3 HMMs $HMM-1_{good}$, $HMM-1_{medium}$ and $HMM-1_{bad}$ is shown in Figure 4.3. According to this test $HMM-1_{good}$ performed exactly as in the first test (as it should since it had the same training data). $HMM-1_{bad}$ also performed much better this time. The detected state transition point to the bad state was recorded at point 80205 which is exactly the place where the first recorded values over 400 took place. However, $HMM-1_{medium}$ performed poorly. The first point were $HMM-1_{medium}$ had the highest log-likelihood was at the 77727th observation. However, the first recorded value which was less than 300 was at the 77029th observation.

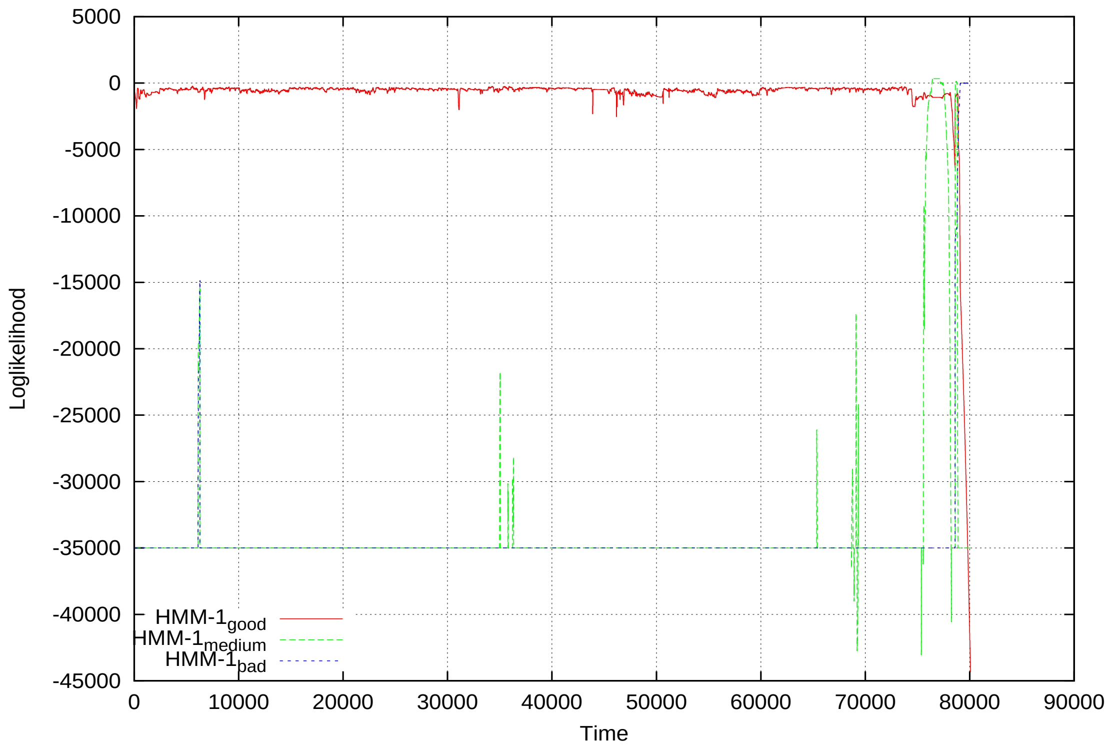

Figure 4.3: Log-likelihood values when 3 HMMs were trained to perform state detection. Reported log-likelihood values of -Infinity has been substituted with -35000 to reduce cluttering in the graph.

From the results of the second state detection test it seems as though more care has to be made when dividing training data between HMMs. Even though $HMM-1_{bad}$ managed to detect the exact point where values over 400 were first recorded, the error that $HMM-1_{bad}$ had from the first prediction test seems to have been shifted to $HMM-1_{medium}$. Perhaps introducing even more HMMs to represent even more health states might help to reduce the state detection error.

The reason that $HMM-1_{bad}$ performed so much better in the second test is the reduced variation in training data. During the first test $HMM-1_{bad}$ had to be able to recognize values ranging from 290 to 512. However, in the second test $HMM-1_{bad}$ only had to recognize values higher than 400.

As a last test to see if the state detection would become better if the variation in data values was reduced, we standardized the data to reduce the variance between values. However, this only reduced overall state detection accuracy since the data became too similar.

For this particular dataset, it might also be the case that current state detection on such datasets with huge variations in measured values, as the electric motor data, might be performed better with just a single HMM which have been trained on the entire dataset. However, this option has not been tested due to time constraints.

4.3 Future state prediction

When the future state prediction was tested, only the generated data was used. The prediction method presented in[5] was tested for two different observation vectors. One vector was similar to the training data, while the other vector was designed to be longer and have more time between state transition points. The test for the future state prediction is how well it manages to estimate the final transition point.

The state transition points for the first test vector is shown in Figure 4.4 and was obtained by using the current state detection method. The state transition points of the first test vector was as follows:

- First state transition point $t_{good \rightarrow medium-good}$ = 289.
- Second state transition point $t_{medium-good \rightarrow medium-bad}$ = 1367.
- Third (and final) state transition point $t_{medium-bad \rightarrow bad}$ = 2607.

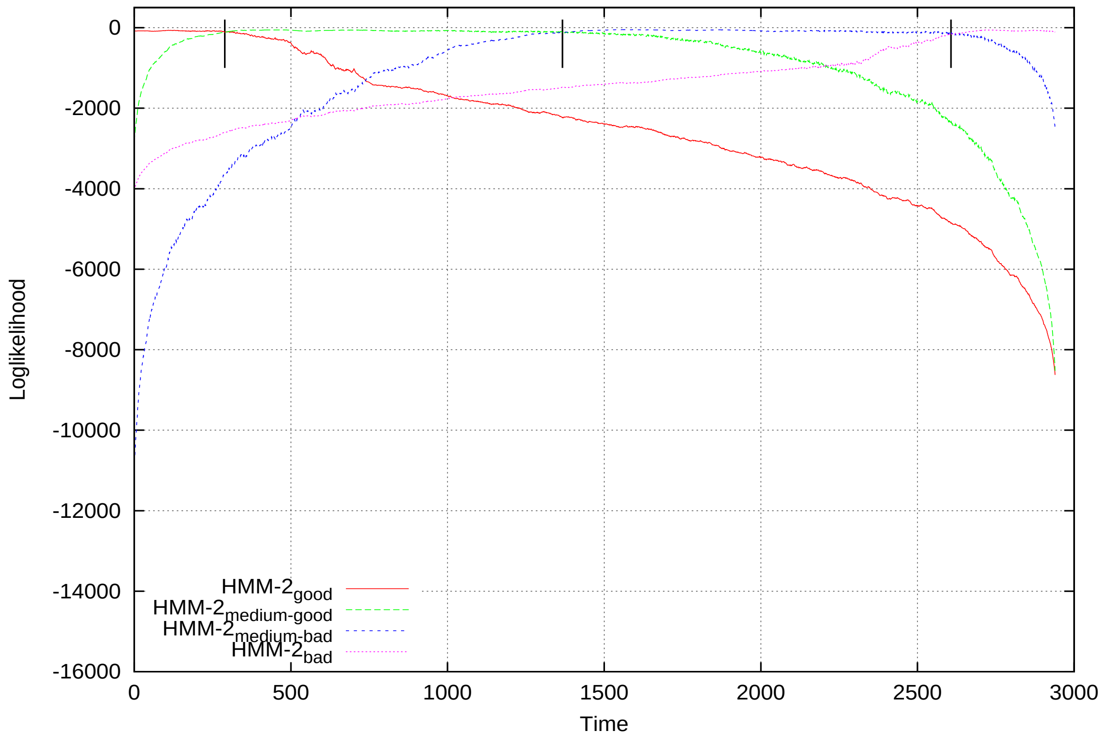

Figure 4.4: State transition points for the first vector used for future state prediction.

Once the first state transition point was recorded it was possible to make a prediction about the final state transition point. The first possible state transition prediction is shown in Figure 4.5. The figure shows the first recorded transition point as well as the predicted final state transition point ($t_{medium-bad \rightarrow bad}$). The prediction will be the most inaccurate of the two predictions as it only takes one previous state transition into account. The first prediction estimated that the final state transition point would be at observation number 2022, while the actual state transition occurred at observation number 2607.

It is not expected for any of the final state transition predictions to be able to exactly estimate the correct state transition points. However, the early prediction gives a rough estimate as to when the final state transition will occur, and it should only be interpreted as an rough estimate of the remaining useful life of the component

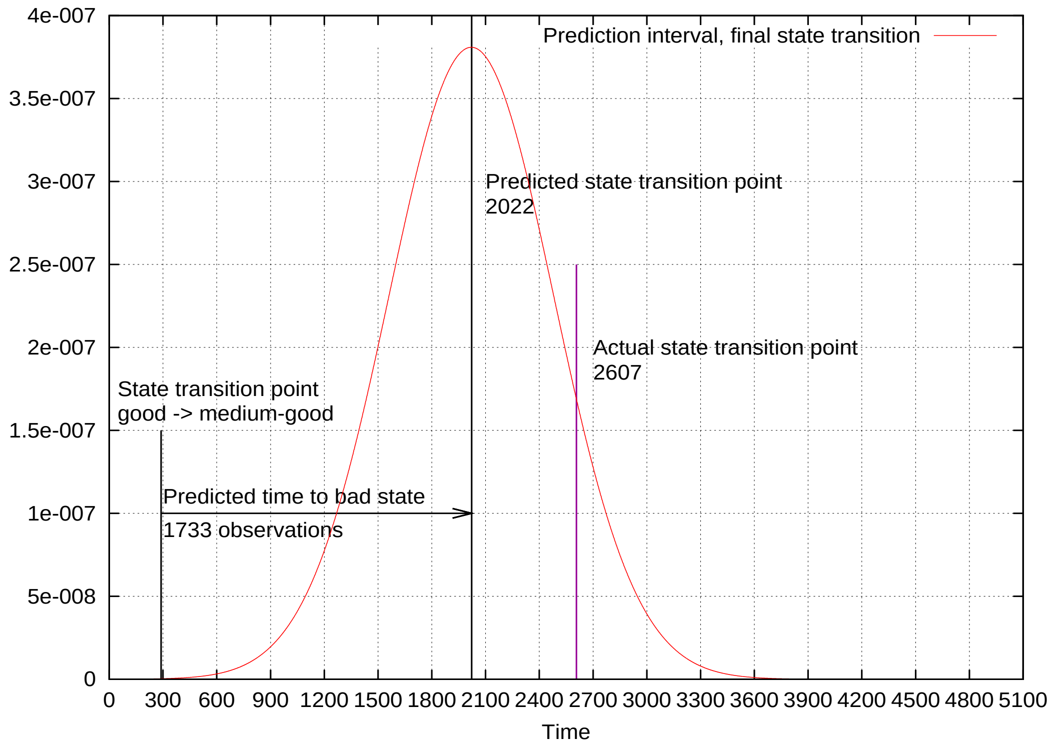

Figure 4.5: The estimated state transition interval when the first state transition is detected.

After the second state transition point has been detected it is possible to make a more accurate prediction about the final state transition point of the component. Figure 4.6 shows the final state transition estimate when both $t_{good \to medium-good}$ and $t_{medium-good \to medium-bad}$ have been recorded. In this case the prediction is more accurate and it is clear that more information about previous state transition points will increase the prediction accuracy.

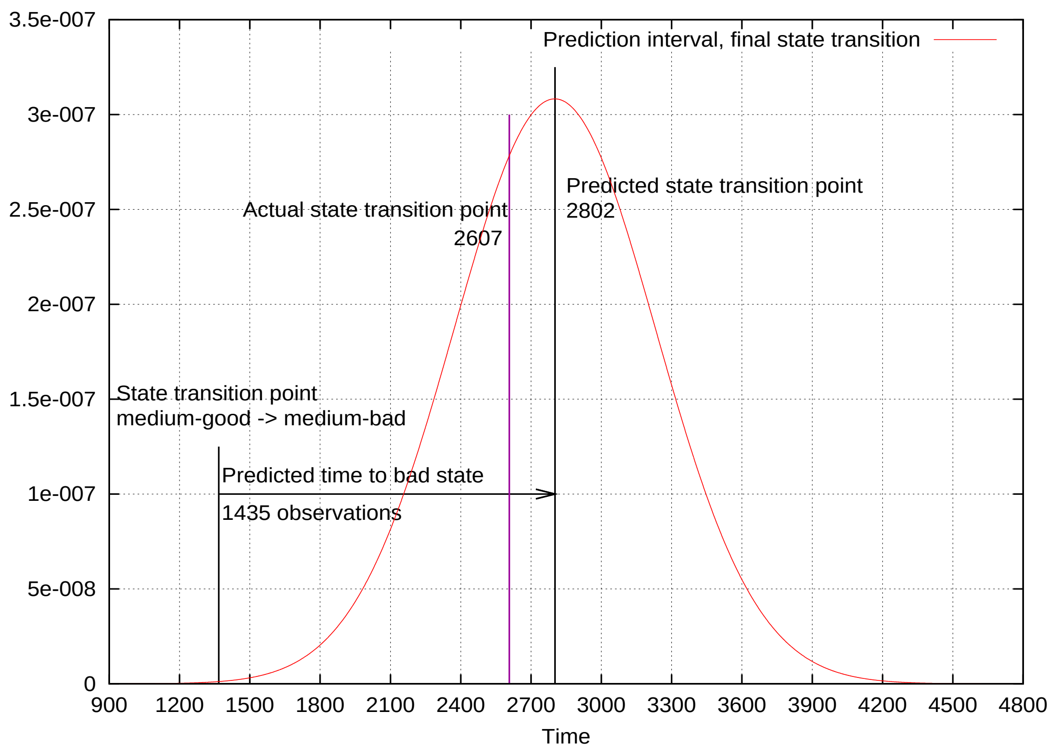

Figure 4.6: The estimated state transition interval when the second state transition is detected.

Figure 4.7 shows both prediction graphs and their estimated transition points. It can be seen that the second prediction graph is more focused(i.e. the prediction interval is narrower). However, the bell curve of each graph is still to wide to offer much meaningful information about the final state transition point. With access to more training data the prediction intervals, represented by the bell curves in Figure 4.7, may change to be narrower and thus offer meaningful information about the final state prediction. For example if it could be tested and proved that most observations would fall within one standard deviation of the estimated mean it would be possible to narrow down the expected state transition point even more.

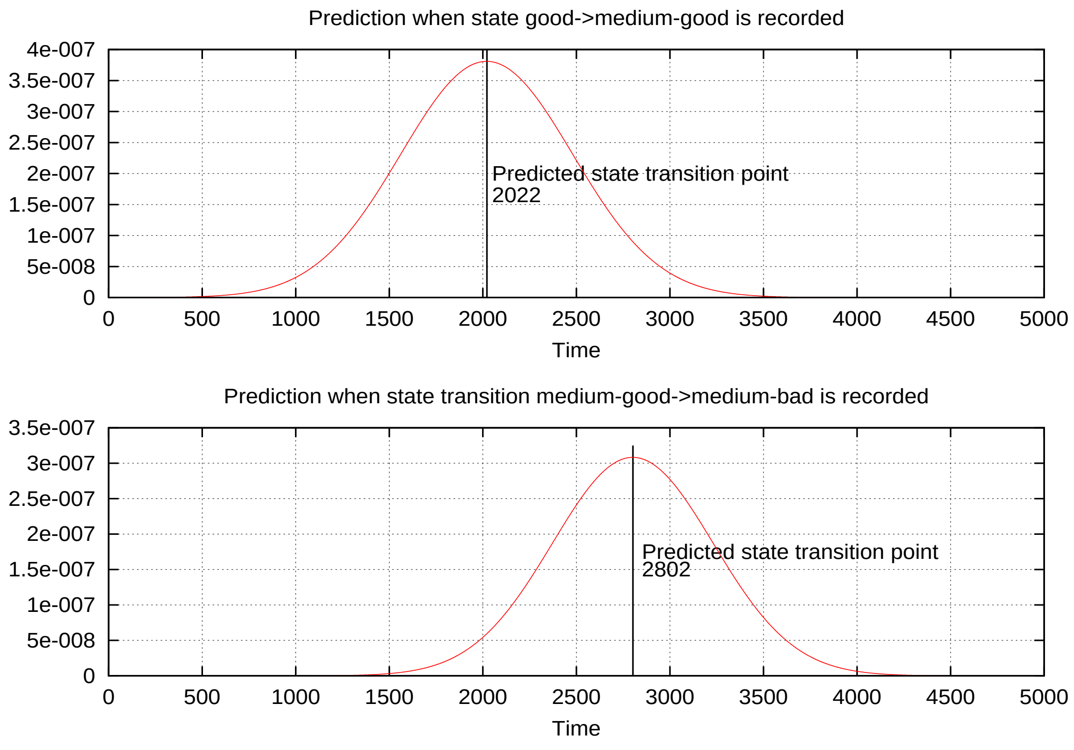

Figure 4.7: Comparison of the two different state transition distributions.

The second test vector was designed to spend more time between each state transition. This was done to test the prediction method against an observation which was radically different from the data used to estimate the initial state prediction parameters. The state transition points are shown in Figure 4.8. The state transition points are also shown in the following list:

- First state transition point $t_{good \to medium-good}$ = 620.
- Second state transition point $t_{medium-good \to medium-bad}$ = 2808.
- Third (and final) state transition point $t_{medium-bad \to bad}$ = 5243.

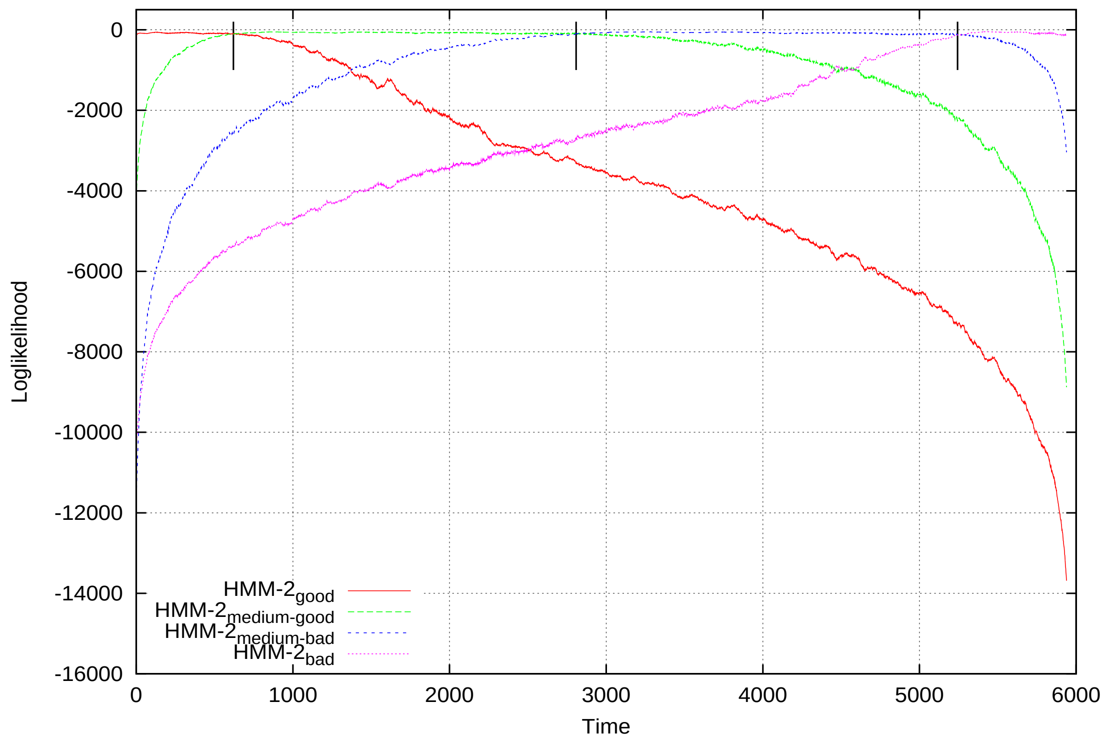

Figure 4.8: State transition points for the long future state prediction vector.

When performing final state prediction on second test vector, the state transition estimation procedure did not even come close to predicting the final state transition point. It is clear that the final state transition estimation relies heavily on the state transition points offered by the training data.

The prediction which only relied on the first transition point $t_{good \rightarrow medium-good}$ is shown in the top half of Figure 4.9. In this case the predicted final state transition point was 3429, not even close to the actual value of transition point of 5243.

The prediction which relied on both $t_{good \rightarrow medium-good}$ and $t_{medium-good \rightarrow medium-bad}$ did not make a more accurate prediction in this case. The estimated state transition point and corresponding bell curve is shown in the bottom half of Figure 4.9. The predicted final state transition point in this case was 3392, which is far from the actual state transition point of 5243.

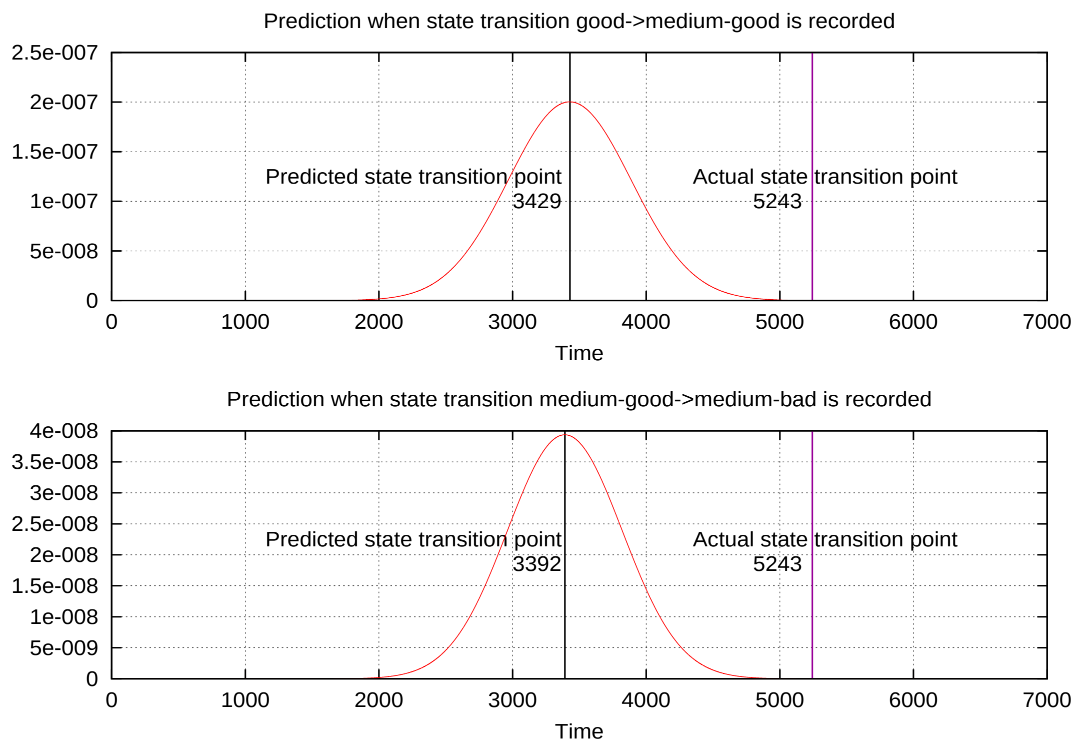

Figure 4.9: Comparison of the different prediction distributions for the long future state prediction vector.

Based on the two different prediction tests it is possible to say that the chosen method of estimating state transition points require a lot of varied training data, to be able to produce somewhat accurate results for observation vectors of different length.

Chapter 5

Summary, conclusion and contribution

In this chapter we will present a short summary of all the work that has been performed, review the results of our tests, and draw the final conclusion of our work. We will also write about the contribution this work can bring to the field of Condition Based Maintenance and hidden Markov Model.

5.1 Work summary

During this the work on this thesis several hidden Markov models (HMM)) have been trained and tested in performing different tasks related to Condition Based Maintenance (CBM). All tests that have been performed have made use of a minimum of training data and expert knowledge about the system which the data represents.

The first test tested how well a HMM could detect anomalies in system measurements. The second test was determining how well two HMMs could detect the current health state of an electric motor. The third and final test tried to determine how well HMMs could estimate when a system was going to fail.

5.2 Results

In this section we will take a brief look at the results from all tests that were performed.

5.2.1 Anomaly detection

A single HMM was presented with temperature measurements which were taken from an electric motor. The HMM would respond to the presented observations by giving a log-likelihood estimate of how well the measurements were recognized by the HMM. Based on the log-likelihood values it was possible to detect anomalies that occurred in the data.

As long as it is possible to determine a certain log-likelihood value that would mean that an observation does not belong to a given HMM, it is entirely reasonable to say that HMMs are able to analyse data and detect anomalies that may occur. By combining the anomaly detection with expert knowledge about the system it is possible to define errors that have occurred. When errors are found and the reason behind the error is established it is possible to train new HMMs that can detect these errors in future measurements.

Anomaly detection will usually require less training data since so few states have to be identified. However, even though less training data is required there is a greater demand for access to expert knowledge about anomalies that are detected.

5.2.2 Current state detection

During this thesis two HMMs were trained to detect two different health states in the electric motor. If expert knowledge had been available then it might have been possible to define more error states and train more HMMs to better classify the current health-state of the system.

By dividing the system into only two health states one of the HMMs was given data that would make it able to determine the good-health-state ($HMM-1_{good}$) with good accuracy. However, the HMM that was going to recognize data that would indicate that the system was in the bad-health-state ($HMM-1_{bad}$) did not manage to recognize the data in a good manner. The reason for this is that the huge variation in values in the training data made it difficult for $HMM-1_{bad}$ to be properly trained to recognize the data. By reducing the amount of training data to only include measurements which had similar values it was possible to make $HMM-1_{bad}$ detect errors more accurately.

When the training data was reduced for $HMM-1_{bad}$, a completely new HMM had to be trained to attempt to recognize the data which was removed from $HMM-1_{bad}$s training set. The result of this was that the low detection accuracy which had previously existed in $HMM-1_{bad}$ was now moved over to the new HMM ($HMM-1_{medium}$).

It is possible that the low classification accuracy could be eliminated completely by training more and more HMMs. However, by training many HMMs one would end up with many HMM which may not be able to tell something useful about the health-state of the system.

The form of the data used for this test could indicate that it would have been better to only train a single HMM on all the available training data. However, since the single HMM approach has not been tested, it is impossible for us to say wether this would have worked properly or not. It is possible that it would have been difficult to properly train a single HMM to recognize the different health states in an accurate manner. However, without performing any tests no conclusion can be drawn on the current state recognition of a single HMM.

The approach of using several HMMs to recognize different health states in the electric motor data seems to be unsuccessful. Using multiple HMMs to detect health states seem to work best when the data shows a more gradual approach to-wards system failure.

5.2.3 Future state prediction

The future state prediction presented in this thesis attempts to predict the state transition points of a component. The HMMs and probability distributions which were necessary for state transition prediction were created from only two training vectors worth of observations. When creating this system for state transition point prediction two different training vectors is the minimum that can be used.

The prediction was tested on two different observation vectors containing a different number of observations. The first prediction test vector contained 3000 observation (the training vectors also had 3000 observations each), while the second prediction test vector contained 5000 observations.

The prediction on the first test vector was shown to be reasonably accurate. The final state transition point could be estimated to be within 200 observations of the actual state transition point.

The prediction on the second test vector could not even come reasonably close to an accurate prediction. The best prediction was that the final state transition would occur at observation number 3429 while the actual final state transition point was at observation number 5243.

The ability to predict state transition points seems to be heavily restricted by the amount of available training data, and the variation of state transition points contained in the training data. If each component can be expected to last for about the same amount of time then this prediction method might work well. However, if a new component has an extremely long lifetime then the prediction system will have difficulties handling this.

If the generated data had been partitioned to include more health-states, then the prediction might also have worked for the second test vector. This is due to the fact that the state transition prediction becomes more accurate when more information is used from the component being analyzed (i.e. more state transition points are used as part of the prediction).

If the observations are made at irregular intervals, then the state transition point estimation may also become more inaccurate. This is due to the fact that the estimated transition point is represented as an observation number were the state transition is likely to take place. If the time from current-observation to estimated transition point can not be accurately estimated, then the estimated transition point will become even more inaccurate.

5.3 Conclusion

During work on this master thesis we have created several HMMs and tested them in their ability to perform several tasks related to CBM. These tasks included anomaly detection, current state detection and future state prediction.

Using HMMs for anomaly detection has been shown to work well. Together with expert knowledge about a mechanical system it should be possible to identify anomalies in measured data, and classify new errors which the system might be able to detect at a later time.

The proposed approach to current state detection has been shown to have some weaknesses when used on data that has sudden, large changes in value. A single HMM approach might be more suitable for use on this kind of data. Alternatively, if more training data could be gathered to better train the HMMs to recognize the different health-states represented in the data, then the multi HMM approach might also be successful.

The examined method of future state prediction has been shown to not perform so well when little training data exists. However, if a lot of training data is present, and the data can be divided into many different health-states then the examined prediction method might also be viable. In the context of this thesis, were one of the objectives was to use little training data, we would say that the examined prediction method is not viable if the lifetime of different components can be expected to vary a lot.

5.4 Contribution

We have demonstrated a way of using HMMs in the context of different CBM problems. While working on this thesis we have tested the HMMs ability to detect measurements that differ from the previously seen values (anomaly detection), classify different measurements (in the form of health states), and predict future development of a system. All this has been tested with a minimum of available training data.

We have shown that HMMs can be used to analyze sensor data, classify wether the data is known or unknown, and predict further development based on previously observed data. Hopefully this work can contribute to further research into the possible uses of hidden Markov models and their uses and limitations for analysing different kinds of data.

Bibliography

[1] Description of corrective maintenance. http://www.wisegeek.com/what-is-corrective-maintenance.htm.

[2] Overfitting. Online article describing overfitting. http://www.viswiki.com/en/Overfitting.

[3] Wikipedia article on condition based maintenance, 6 january 2010. http://en.wikipedia.org/wiki/Condition-based_maintenance.

[4] Wikipedia article on hidden markov models. 6 january 2010. http://en.wikipedia.org/wiki/Hidden_Markov_model.

[5] P. Baruah and R. B. Chinnam. Hmms for diagnostics and prognostics in machining processes, 15 March 2005.

[6] Dan McCarthy Carey Bunks and Tarik Al-Ani. Condition-based maintenance of machines using hidden markov models, 27 March 2000.

[7] 2007 Industrial Accident Prevention Association. Description of preventive maintenance. http://www.iapa.ca/pdf/prevent.pdf.

[8] Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis, second edition*. Prentice-Hall, Inc. A Division of Simon & Custer Inc. Englewood Cliffs, New Jersey 07632, 1988.

[9] Kevin Murphy. Hidden markov model (hmm) toolbox for matlab. http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm.html.

[10] Kevin Murphy. Hidden markov model (hmm) toolbox for matlab documentation. http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm_usage.html.

[11] Chris Lynn Doug Goodman Neil Kunst, Justin Judkins. Damage propagation analysis methodology for electromechanical actuator prognostics. *Aerospace conference, 2009 IEEE, Print ISBN: 978-1-4244-2621-8*, 2009.

[12] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition, February 1989. Proceedings of the IEEE, VOL 77, NO. 2.

[13] M.S. Ryan and G.R. Nudd. The viterbi algorithm. Department of Computer Science, University of Warwick, Coventry. CV4 7AL, England.

[14] Narada Warakagoda. The baum welch algorithm. http://jedlik.phy.bme.hu/˜gerjanos/HMM/node11.html#SECTION00243110000000000000.

[15] Narada Warakagoda. Definition of hidden markov model, May 1996. http://jedlik.phy.bme.hu/ gerjanos/HMM/node4.html.

[16] Eric W. Weisstein. Markov process. from mathworld–a wolfram web resource. http://mathworld.wolfram.com/MarkovProcess.htmls.

[17] Chiman Kwan Steven Y. Liang Qiulin Xie Leonard Haynes Xiaodong Zhang, Roger Zu. An integrated approach to bearing fault diagnostics and prognostics. *American Control Conference, 2005. Proceedings of the 2005*, 4:2750–2755, 2005.