Sensor Based Dairy Cow Estrus Detection

Miika Ihonen

School of Electrical Engineering

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

Espoo 16.1.2015

Thesis supervisor:

Prof. Arto Visala

Thesis advisor:

M.Sc. (tech.) Samuli Mäkinen

Aalto University
School of Electrical Engineering

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Miika Ihonen

Työn nimi: Sensor Based Dairy Cow Estrus Detection

Päivämäärä: 16.1.2015 Kieli: Englanti Sivumäärä: 7+66

Department of Automations and Systems Technology

Professuuri: Jotakin kivaa

Työn valvoja: Prof. Arto Visala

Työn ohjaaja: M.Sc. (tech.) Samuli Mäkinen

Tässä opinnäytetyössä tutkitaan sensoreihin persutuvia lypsylehmän kiimantunnistusmentelmiä. Tutkimusta varten kerättiin kiihtyvyys- ja lämpöanturidataa lehmiltä kaulalle kinntityillä datan keruulaitteilla. Työn lopputuloksena on kolme erilaista kiihtyvyysanturin dataan perustuvaa lgoritmia. Kaikki algoritmit suoritutuivat kiimantunnistuksesta. Kuitenkin passiivisuuden tunnistukseen perustuva algoritmi osottautui kaikista luotettavimmaksi ja varmimmaksi erilaisilla parametreilla. Työn johtopäätöksenä on, että lypsylehmän kiima on tunnistettavissa kiihtyvyysanturiin perustuvalla sensorilaitteella. Kuitenkin varmojen johtopäätösten tekemiseksi, tämän työn tulokset tulisi vielä vahvistaa onnistuneilla siemennyksillä.

Avainsanat: Avainsanoiksi valitaan kirjoituksen sisältöä keskeisesti kuvaavia käsitteitä

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Miika Ihonen

Title: Thesis template

Date: 16.1.2015 Language: English Number of pages: 7+66

Department of Radio Science and Technology

Professorship: Circuit theory

Supervisor: Prof. Arto Visala

Advisor: M.Sc. Samuli Mäkinen

This research studies sensor based dairy cow estrus detection. For the study, we recorded motion and temperature data with a collar sensor. The data was used in algorithm development end evaluation. In result, we developed three different algorithms, all suitable for micro-controller devices. All the developed algorithms succeeded in the estrus detection against a reference system. However, inactivity detection based algorithm was the most reliable and tolerant to different configurations. In conclusion, the estrus is detectable with accelerometer based sensors. However, in order to make secure conclusions, the results of this study should be verified by a successful insemination.

Keywords: Resistor, Resistance,
Temperature

Prologue

I would like to thank you everyone. [40] [11]

Otaniemi, 16.1.2017

Miika S. Ihonen

Contents

Symbols and Abbreviations

Symbols

g	acceleration unit $\approx 9.81[\mathrm{m/s^2}]$
$\mathbf{B}$	magneettivuon tiheys
c	valon nopeus tyhjössä $\approx 3 \times 10^8[\mathrm{m/s}]$
ω_{D}	Debye-taajuus
ω_{latt}	hilan keskimääräinen fononitaajuus
$\uparrow$	elektronin spinin suunta ylöspäin
$\downarrow$	elektronin spinin suunta alaspäin

Operators

$\nabla \times \mathbf{A}$	vektorin $\mathbf{A}$ roottori
$\dfrac{\mathrm{d}}{\mathrm{d}t}$	derivaatta muuttujan t suhteen
$\dfrac{\partial}{\partial t}$	osittaisderivaatta muuttujan t suhteen
$\sum_i$	summa indeksin i yli
$\mathbf{A} \cdot \mathbf{B}$	vektorien $\mathbf{A}$ ja $\mathbf{B}$ pistetulo

Abbreviations

MCU	Micro-Controller Unit (or micro-controller)
SD	Secure Digital
SDHC	High Capacity Secure Digital
SDXC	Extended Capacity Secure Digital
SPI	Serial Peripheral Interface bus
I^2C	Inter-Integrated Circuit bus (also IIC)
EEPROM	Erasable Programmable Read-Only Memory
SRAM	Static Random-Access Memory
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
FIFO	First In, First Out
IDE	Integrated Development Environment
FFT	Fast Fourier Transform
RISC	Reduced Instruction Set Computer
CISC	Complex Instruction Set Computer
USB	Universal Serial Bus
ISR	Interrupt Service Routine
I/O	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit

1 Introduction

People have been using milk as a source of nutrition since the beginning of animal husbandry. The research show that the drinking of milk started 8000 years ago in the present Turkey area. Ever since the dairy farming has spread all over the world. Meanwhile, the spectrum of the available dairy products and fabrication techniques have been increasing. Initially, the headcount of a cattle was small, and it served only a family or a local community. However, since industrialization the farm sizes increased whereas, the number of farms begun to decrease. Same trend has been going on thereafter. Nowadays, strict competition in local and global markets has driven the farming financial in challenges. Consequently, human labor has become an expensive recourse cutting the profitability. Moreover, it has caused the increase of the workload of the farmers. Thus, they have less time to spend with the cattle observing the status of estrus and health issues. Punctual estrus detection is fundamental factor in keeping the calving interval within the optimal range. Extended calving intervals have direct affect in milk yield and the profitability. In Finland, the average calving interval has already exceeded 400 days whereas, 360 days is the optimal and recommended interval. Traditionally, estrus and health issues have been detected sight-wise by a cattle tender. This method is time consuming and considered inefficient. However, such symptoms as lameness are still difficult to detect with sensor-based solutions. Visually, lameness is rather easy to detect. Nevertheless, sensor-based technologies aim to detect health issues before they escalate into sever conditions such as lameness.

Currently, there are numerous technological solutions available as an alternative to human labor. However, they are relatively expensive investments and their actual payback time is difficult to define. Therefore, specially many smaller farms have postponed the use of modern technological aids in cattle monitoring. Additionally, even large farms in developing countries are in the same stage and still rely on human labor and traditional methods. In addition to the high cost and long payback time, the functionalities and the performance level of the solutions can vary significantly. Thus, achieving of complementary solutions are rather challenging. Typically, these commercial solutions offer aids for estrus detection and general health tracking. In fact, they tend to trigger alarms for cattle tender to start inspections instead of providing accurate health status.

Based on this background information, the rigorous detection of estrus for insemination is the most critical issue in nowadays dairy farming. Thus, this study aims to develop and evaluate an effective algorithm for dairy cow estrus detection. Conventionally, cattle tender monitors cattle and detects estrus behavior. That is, a cow being mounted by other cows is considered as secure indication of ongoing estrus. There are additional behaviors giving a hint of ongoing estrus. However, confident detection requires several occasions of these behaviors. By contrast, in sensor-based estrus detection it is more convenient to detect proestrus instead of the actual estrus. The use of accelerometer as the fundamental sensor in these applications is reasonable, hence, the behavior of the cow is very active in this phase. Additionally, detection of proestrus provides the farmer enough time to prepare the insemination before the actual estrus. Unfortunately, cows in tie-stall are not able to be active walking around the cowshed. Thus, only loose-housed cows are considered in this study. Additionally, the results of this study could be adoptable for beef cattle as well as other species. Nevertheless, these options are not discussed in this study. In addition to the active behavior of the cow, its body temperature rises during the estrus. Therefore, some commercial solutions include body temperature measurement. Similarly, the hardware solution in this study includes a temperature sensor. Nevertheless, the accelerometer is the fundamental component in estrus detection and the temperature sensor is to confirm or question the detection.

The focus of the evaluation is in reliability and punctuality. Therefore, the algorithm shall not trigger false estrus or miss a true estrus either. Additionally, the timing shall remain within reasonable tolerance. Thus, the resulting algorithms shall be able to detect estrus in real-time. In addition to these requirements, the algorithm shall be stand-alone solutions suitable for low-cost micro-controller devices. Therefore, in this study we use common low-cost components in the sensor device hardware configurations. Additionally, the hardware design is rather simple in order to keep the costs low. However, in wearable battery-based solutions the high capacity batteries expensive and their cost may exceed the price of the other hardware. Thus, low energy consumption is considered in the discussions of this study. In addition to the estrus detection, we attempt to review the possibility of behavior monitoring as well. However, the behavior monitoring is a minor topic of this study. Therefore, the results related to behavior monitoring only are discussed only briefly in this study.

Efficient data analysis, algorithm development and testing are an absurd approach. Specially, when the estrus cycle of a cow lasts approximately 21 days and the phases of proestrus and estrus several hours. Therefore, it is essential to use recorded sensor data in algorithm development and in testing as well. Consequently, implementation of suitable data recording software is one of the core topics in this study. Furthermore, properly implemented data recording application gains benefits in possible future studies as well. The data recording application shall be implemented on similar cost-efficient hardware platform as the actual estrus detection algorithm would be. In contrast to software, designing of hardware is not included in this study. However, we will introduce the hardware platform used in this study. Furthermore, we will discuss of its components and their functionalities in reasonable detail. In despite of the requirements of real-time punctuality of the estrus detection algorithms, real-time algorithm testing is excluded from this study. By contrast, the algorithms shall be implemented and tested with appropriate software tool and the results are discussed in the results section. Nevertheless, the convertibility of the algorithms are considered throughout the development process. Furthermore, the actual algorithm implementation for a micro-controller are not discussed in this study.

In despite of the real-time punctuality of the resulting algorithms, testing of the algorithms on-site is not in scope of this study. Conversely, the algorithms shall be developed and tested with a suitable computer software. The algorithm development requires real cow data from a dairy farm. Therefore, implementation of a suitable data recording software for a low-cost micro-controller device is one of the core topics in this study.

Kerro menetelmistä.

Onko muita rajauksia?

Puuttuuko jotain muita perusteluja tutkimuksen tekemiselle, valinnoille yms?

2 Background ok

Originally, humans were hunter-gatherers who obtained food by collecting plants and pursuing wild animals. The methods for acquiring food have changed substantially since the beginning of the agriculture. That is, plants and animals are grown centralized in large farms instead of numerous small producers. Actually, the animal husbandry has been estimated to been started more than 10,000 years ago in western Asia. Accordingly, goats were among the first domesticated animals in human history [39]. Thereafter, people have been domesticating other species e.g. cows, sheep and pigs for milk, meat and other animal products. In addition to the quantity of various species, the headcount has been increaing with the population. Furthermore, the industrial revolution has started a trend of growing farm sizes and a loss of smaller farms. Recently, the total headcount of the world has been estimated almost up to 1,000 million heads in 2016 [3].

Nowadays, the cattle breeding is divided into two trends of beef and dairy farming. As a result, the breeds of beef cattle and dairy cattle are considerably different in physics and by nature. The beef breeds are more muscular whereas milk breeds are more tame. Moreover, the scope of beef breeding is in rapid brawn growth, while high milk yield is the target of the dairy cattle breeding. In spite of the same origin of the breeds, they are as distinct as different species. Therefore, the scope of this study is only in dairy farming. Accordingly, the following subsections will discuss briefly of the history and the basics of the modern dairy farming 2.1. Furthermore, we will survey through the life of a cow 2.1.1 and discuss of related issues. Additionally, we will focus in two fundamental cycles in the life of a dairy cow, estrus cycle and lactation cycle 2.1.2. These cycles are directly related to the milk yield and profitability of the dairy farm. In addition to the discussions of the dairy cow, we will take an overview on dairy cow monitoring 2.2.1. The overview includes as well as traditional methods as currently available technological aids. Lastly in the background section, we will survey through the most recent research and studies in dairy cow monitoring 2.2.2. In the survey, we will focus specially in wearable sensor devices and behavior monitoring.

2.1 Dairy Farming ok

As discussed in previously, the origins of animal husbandry are over 10,000 years old. Whereas, the drinking of milk started 8,000 years ago in present Turkey area. Thousand years later, dairy farming started to spread into Europe and thousend yars after that into Africa [39]. Thereafter, dairy farming has spread all over the world. Meanwhile, the variety of dairy products has exploded simultaneously. In result, there are numerous different milk, cheese and yogurt products as well as other dairy refinements. Additionally, dairy products have an essential part in human nutrition nowadays. Furthermore, milk and other dairy products are produced more than ever before. Concurrently, the farm headcounts have been increasing whereas the number of farms has started to decrease. Alone in Finland, there were total of 909,000 cows of which 282,000 milking cows. Meanwhile, 7890 farms delivered milk to milk processing plants in the end of the year 2015. As aresult, the average yield of a farm was 279 thousand liters whereas the average yield of a cow was 8300 liters. Accordingly, the total milk yield in Finland was 2365 million liters in 2015 [22].

Figure 1: Cows in a tie-stall-cowshed [15]. Cows are tied in stall and they are not able to move freely. They also have less space than in a loose-housed cowshed.

Correspondingly in the Unites States of America, there were more than 89 million cows and calves of which more than 9 million were milking cows in 2014. The corresponding cash income of the dairy farms was more than 49,349 million dollars in 2014 in the USA [2].

Originally, cows were wild pasture animals and, afterwards, domesticated by humans. At the beginning of animal husbandry, the people was migrating nomads. Therefore, the cattle traveled with people. Eventually, people started farming and settled in constant regions with their animals. First, the animals were held in yards but then people started to build structures for protection and easier keep of the animals. Next, people started to keep their cattle inside buildings. In linear pottery culture, people and animals lived together inside longhouses. Finally, people started to build separate buildings for theri animals. Special building for keeping cows are called cowsheds. Currently, the cowsheds are divided in two types of tie-stall and loose-housed cowsheds [18]. In general, tie-stall cowsheds are smaller and, therefore, tighter than loose-housed cowsheds. Furthermore, cows are not allowed to move freely in tie-stall. Conversely, In loose-housed cowshed cows are allowed to move freely round the clock. Additionally, they may have free access to pasture in some solutions. Recently, the tie-stall cowsheds have been under critique. That is, the cows are not able to behave as social animals. Additionally, the monitoring and health keeping of the cows is more difficult in tie-stalls. Therefore, most of the new-builds are rather loose-housed than tie-stall cowsheds. In addition, it is considered the cows of being happier and healthier in loose-housed cowsheds.

Figure 2: Cows in a loose-housed-cowshed [32]. Cows are able to move freely and act as social animals.

2.1.1 Dairy Cow

Tähän alkuun vielä jotakin lehmän perustietoja, kuten arvio rotujen määristä. Lehmän keskimääräinen paino, koskeus, pituus jne. Ehkä myös maininta siitä, kuinka paljon lehmä tarvitsee tilaa mm. laskeutumiseen ja ylösnousuun.

Previously, we surveyed through the history of animal husbandry and discussed of the beginning milk producing. Additionally, we introduced such cowsheds as tie-stall and loose-housed cowsheds. Correspondingly, this subsection will debate on the dairy cow itself in general level. Whereas, the subsequent subsections will focus on such milk yield related cycles as estrus and lactation cycles. Inherently, cows are plain and herd animals. Moreover, they live in hierarchy [13]. In large herds they form smaller groups where they do their daily activities such as eat and rest together [13] [18].

Typically, cows lay down approximately from $11\,\mathrm{h}$ to $12\,\mathrm{h}$ every day. Meanwhile, they stand up and change their pose several times [43]. Additionally, they may move their location between haunts and watering places. Thus, in loose-housed cowshed a cow may walk from $400\,\mathrm{m}$ to $800\,\mathrm{m}$ per day [31]. On pasture, their daily walking range may extend to several kilometers [31] [13]. However, cows are very cautious animals. Thus, insecure circumstances such as slippery ground or high steps can reduce their daily range. Furthermore, cows can be easily injured in challenging places. Naturally, injuries effects to their health and consequently to profitability [18]. Nevertheless, Walking enforces the health of them, increases hormonal activity and metabolism [31].

In addition to their normal activity, cows may have exceptional states. Typically, these states become apparent in their behavior. That is, sicknesses and injuries reduces their activity level, whereas, proestrus increases it. Phases of estrus are covered in detail in section 2.1.2.

Tänne vähän lisää tietoa lehmien terveysongelmista, kuten ontumisesta, sorkkahommista ja ruoansulatusvaivoista.

Cow can lick all of its body excluding neck and head. Cows doze standing and sleep lying. The estrus period is approximately 21 days and the estrus lasts from 12 to 16 hours. A bull may detect estrus 2 days before the main estrus. [13]

Access to fresh and clean water is vital. [31]

2.1.2 Lactation and Estrus Cycles

In previous subsections, we discussed of dairy farming and dairy cow in general. Consequently in this section, we will proceed the discussion to lactation cycle. The lactation cycle is emphatically related to the milk yield and thus, the profitability of the farm. The lactation cycle means the period between two calves. Thus, it is also called as calving period. After the calving a cow begins to lactate, which is the the actual purpose of a dairy cow [18]. However, the calves are more or less a secondary product in dairy farming and they are not in scope of this study. Nevertheless, the milk yield if a cow increases in the first weeks after the calving. However, after the first weeks the milk yield begins to decrease as illustrated in picture 3. Consequently, it is not cost-efficient to keep on milking the cow infinitely. Therefore, cyclic calving is preferred in order to maintain the profitability. However, the cow will stop milking before calving which causes a dry period.

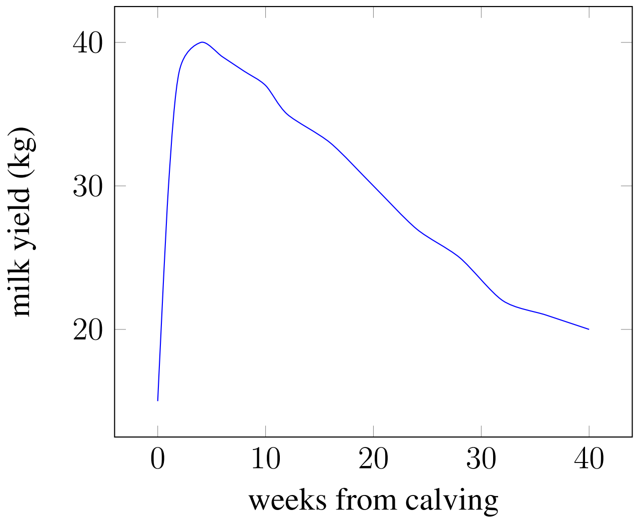

Figure 3: An illustrative lactation curve of a dairy cow representing the realtion between the weeks in milk and the milk yield. After the beginning of the lactation, the milk yield reduces in the function of time [23].

This causes a dilemma. That is, too short as well as too long calving interval reduces the total milk yield and cuts the profitability. Is general, it is recommended to keep the calving interval roughly in 360 days. Moreover, the milk yield increases after each calving. Thus, it supports the idea of regular calving [1].

In recent years, the calving interval has been increasing globally. Alone in Finland, the average calving interval has been extended up to 400 days and it has affected to the milk yield. There has been discussions of the root causes for the extended calving interval. In general, whereas the farm sizes have been increasing the cattle tenders have more work and less time for observing the cows. Thus, it has been more difficult for them to detect any estrus behavior and take actions for insemination. Consequently, this trend has forced the farmers to seek technological aids for punctual estrus detection. Thus, the main scope of this study is in developing of estrus detection algorithms for wearable sensor device. The hardware and software designs are intoruced in section 3 and the results are discussed in 4. Additionally, we will survey through of currently available solutions in section 2.2.1.

However, before proceeding to the research of this study, it is necessary to discuss of the estrus cycle of dairy cow. The estrus cycle is the period between ovulations. Actually, the estrus cycle is considered to begin from the ovulation. The normal duration of a cow estrus cycle is approximately 21±3 days. A bull may detect an estrus already 2 days in advance. The estrus lasts from 12 to 16 hours [13]. The estrus is the time frame when the cow is ready to mate or being inseminated. The estrus is also called as standing heat, hence, the cow allows to being mounted by other cows. Traditionally, being mounted by others has been considered as certain sign of ongoing estrus. Before the estrus, cow has a proestrus period. During the proestrus, cow behaves restlessly and usually it attempts to mount other cows. Normally, cows who are not in estrus do not allow to be mounted. Therefore, these are only attempts and are not considered as a sign of estrus. Typically, the duration of proestrus is from 9 to 18 hours [18].

2.2 Health Monitoring and Estrus Detection

The previous subsection 2.1 discussed of the fundamentals of dairy farming. The discussion started form the history of animal husbandry and ended to study of dairy cow itself. The study of dairy cow included basic knowledge of the cow and its environments. Moreover, we discussed of lactation and estrus cycles and their affect on the milk yield and profitability. Additionally, we briefly surveyed through the most common health issues with dairy cow. In continuation to previous discussions, this subsection discusses of different methods for live-stock monitoring. First of the following subsection 2.2.1 surveys through currently used methods and technologies. Correspondingly, the second subsection 2.2.2 discusses of studies of existing technology as well as recent development projects for future solutions.

2.2.1 Current Solutions

Traditionally, live-stock monitoring has been sight-wise task assigned to the cattle tender [18]. However, this method is not considered to be efficient. In the USA alone, the rate of successfully detected estruses has been estimated below $50\,\%$ in large farms. Additionally, this inefficiency leads to annual loss of 800 million dollars for the milk industry [12]. Naturally, the reliability of the observations depends on several such factors as the experience of the cattle tender, availability of time and the amount of cattle. Additionally, even experienced cattle tender might be erroneous with non-familiar cattle. Furthermore, the head count of farms tend to be increasing whereas the number of cattle tenders remaining the same. Consequently, the tenders have less and less time for purely observing the cattle. Therefore, detection of estrus of health issues has become even more difficult. In despite of the challenges, sight-wise observations by the cattle tender are still common monitoring method in small farms and in developing countries [12].

Nowadays, there are several commercial poruducts available for dairy cow monitoring. Most typically, these products are sensor devices attached with a strap to leg [24, 16] or to collar [17, 16, 25] sensor devices are collar or leg devices mounted with straps. [17] [24]. However, there are even tail-attached sensor [25] and in rumen [26] sensors available. In addition to various attachments, the devices have various applications. Some sensors are specialized in calving detection [25] or digesting monitoring [26] only whereas leg and collar sensors provide more wide range of functionalities. Typically, they monitor the cow motion and activity [24, 17, 16, 25]. Moreover, leg sensors can count steps and detect the pose of the cow (standing, lying and walking). Furthermore, the leg and collar sensors are used in estrus detection. Additionally, some devices are capable of monitoring rumination [17] and eating [16]. Most of the devices are planned for wireless data transmission between the sensor device and a computer of farm server [17] [16] [24] [26]. However, some devices require wired USB data transmission [24]. In despite of the technology or the feature being monitored, all the solutions aim to a healthier cow and, thus, improvements in profitability. Others may alert of possible health issues in advance [17, 16, 24] whereas others are guiding to optimize the feeding and digestion [26]. Furthermore, numerous current solutions provide mobile phone interfaces ans most significant alerts are given in SMS messages or in email [17].

2.2.2 Research and Studies

In former subsection we discussed of currently available technological aids for dairy cow monitoring. Different products provide different functionalities from rumination and digestion monitoring to motion tracking and estrus detection. Similarly in this section, we survey various researches and studies for dairy cow monitoring. In contrast to the previous section, the solutions discussed here are not yet commercially available products. Nevertheless, some studies have been using commercially available devices in their research. In this subsection, our scope will be specially in solutions that are not available in current product range such as behavior monitoring. However, studies of the estrus detection are in our interest as it is the main topic of this study as well.

Most typically in these researches, the sensor devices are based on various accelerometers [27] [42] [19] [30] or pedometers [12]. Additionally, such alternative approaches as intravaginal probe has been tested for estrus detection [5] [4]. In despite most of the solutions are accelerometer-based, their approach to the topic is quite different. The most complex of these algorithm aims to full behavior detection with support vector machines [27]. They used an accelerometer based sensor attached to collar of the cow. Their overall performance of the multi-class model was $78\,\%$ precision. However, The precision with lying down and standing up activities were poor. Correspondingly, another research studied use of simple decision tree instead of support vector machines of hidden Markov Models [42]. In this study, they used various time windows from 1 to 10 minutes. In result, the 1 minute window provided better results than 10 minute window. Additionally, this method was barely equally goon in results as the vector machine algorithms.

Another study used a commercially available leg sensor for estrus detection [19]. From the leg they were able to detect whether the cow was lying, standing or moving. The overall sensitivity in estrus detection was $88.9\,\%$ adn error rate $5.9\,\%$. Also they used 1 minute sample periods. In this study, the detected estruses were controlled by success of insemination. In their conclusions, they suggested this method could only support existing methods and the reliability is not high enough for standalone solution. Yet another research used pedometers for a step-count based estrus detection [12]. In this method they were able to improve the visual estrus detection rate up to $84.2\,\%$. Also in this study, the estrus detection was verified by the success of insemination. Either this solution was standalone, hence the purpose was use it together with the sight-wise estrus detection. In addition to these accelerometer-based behavior and estrus detection, one research studied lameness detection as attaching one accelerometer to each limb [30]. Their ground idea was sample the accelerometer data from each limb at $25\,\mathrm{Hz}$ and use wavelet analysis in order to detect asymmetry, which could be considered as lameness.

In contrast to these more traditional accelerometer-based solution, another researche studied the use of intravaginal probe for estrus detection [4] [5]. The probe was designed to be inserted inter vagina and transmit the temperature and conductivity data wirelessly. The idea for this study was to use non-motion based sensors. Therefore, this solution could be used also in tie-stall cowshed. The research showed some promising results but lacked of reliability. In their sequel study they used two sizes of the intravaginal probes, $160\,\mathrm{mm}$ and $120\,\mathrm{mm}$ probes. However, neither of these were successful. The larger probe caused bleeding during the research, whereas the smaller started to rotate and even ejected the vagina. Nevertheless, they were able to found some correlation between the intravaginal temperature as well as the electrical conductivity and estrus.

In behavior based cow monitoring, the sensor devise attempts to recognize on or more of the following acts:

- *Standing* is a state where the cow stand on all of its legs and stays still.
- *Lying* is state when the cow lies down on it side. Cows spend half of their day lying down.
- *Ruminating* is an act where the cow chews the feed from its rumen in order to digest fibers etc.
- *Feeding* means the act when the cow is eating feed.
- *Normal walking* when the cow is in health and walks normally.
- *Lame walking* when the cow has health issues in its legs or claw and its walking is cautions on asymmetrical.
- *Lying down* means when the cow is standing but changes its state into lying.
- *Standing up* is when the cow was lying but stands up.

. The behavior monitoring provides beneficial information for the farmer about the health status as well as the mix of the feed.

3 Research ok

In previous section 2, we discussed of the essential backgrounds of this study. The discussion started form the beginning of animal husbandry and dairy cow in modern farming. The discussion ended in introduction of currently used methods and survey of recent research and studies. Moreover, the background section focused in milk yield related estrus and lactation cycles in present day farming. In conclusion of the background, there is a certain need for an efficient solutions in dairy cow monitoring. Accordingly, wearable wireless sensor devices are currently the most promising option due their overall performance and availability. Respectively, the target of this study is to develop and evaluate convenient estrus detection algorithms for wearable wireless sensor devices. Therefore, this section discusses of required tools and supportive methods in data recording and algorithm development. Additionally, the data recording software in this study are created from a scratch. Thus, it is necessary to introduce the related software implementations in this study.

In this study, the research section has been divided in three subsections. The first subsections 3.1 discusses of data recording. The data records are essential component in algorithm development and evaluation. In this study we will not use already existing data. Thus, we will implement required software as well as record the data for further employment. The subsection introduces the hardware design for the data recording. The hardware discussion covers the main hardware components in reasonable detail. In addition to the hardware design, we discuss of the communication protocols between the main hardware components. Respectively, we introduce the principle work flow of the data recording software. Whereas, the hardware of this study was already existing, the software is self-implemented. Thus, we will discuss of the implementation more in detail. The second subsection 3.2 introduces a set of data processing methods applied on the recorded data. The set of methods consists of basic statistics as well as various digital signal processing tools. In this study, we utilize only the basic statistical analysis and they are introduced briefly. In contrast, the digital signal processing tools are covered in more detailed discussion. Lastly in this section 3.3, we develop three different algorithms for dairy cow estrus detection. All these algorithms have their base in the discussed methods. However, each of the algorithms have their own approach to estrus detection. Nevertheless, all the algorithms detects rather the pro-estrus than the actual estrus 2.1.2. However, the end of pro-estrus indicates the beginning of the estrus. Therefore, this kind of approach is highly applicable. Additionally, it provides the cattle tender more time to prepare for the required actions.

3.1 Data Recording ok

Above, we described the research section of this study in general level. As already stated, this subsection will discuss of the data recording process. The data records are essential components in algorithm development and evaluation. The discussions in this subsection will start from the introduction of data recording hardware. The hardware design in this study is an existing hardware setup. Therefore, we will discuss of the main components and functionalities only superficially. Next, we will discuss of the data recording software.

The software for this study is created from a scratch. Therefore, the software design is discussed particularly. In this study, we recorded data in three different occasions of which, the first occasion with a different software implementation. Consequently, we introduce two different software implementation. Lastly in the data recording subsection, we will describe the actual data recording procedure of all the occasions.

3.1.1 Hardware ok

As stated previously, the data recording hardware in this study is an already existing prototype of a dairy cow sensor device. Originally, the hardware consisted of micro-controller unit (MCU), accelerometer and thermometer. However, it lacked of large enough storage memory for data recording. Thus, the hardware was enhanced with a Secure Digital (SD) memory card slot and with a SD memory card for this study. Otherwise, the hardware design remained the same during the data recording processes. In spite of the mentioned customization, the hardware design is not in the scope of this study. Therefore, the following is rather a description of the hardware than a comprehensive discussion of the design itself. In addition to hardware components, we briefly discuss of the communication between the components. In general, the main hardware components brief descriptions are as follows.

- *Atmel ATmega32u4* is a high perforamance low power 8-bit micro-controller. It is designed for optimizing power consumption versus processing performance. It contains 135 powerful Reduced Instruction Set Computer (RISC) architecture instructions of which most executable in a single clock cycle. [10]

- *Bosch Sensortec BMA222E* is an accelerometer with on-chip motion triggered interrupt controller. Thus, it enables motion-based applications even without utilizing a micro-controller. It is capable of measuring acceleration in three perpendicular axises. BMA222E is designed for various consumer products from game controllers to pedometers. It is small sized and low power consuming. Therefore, it is suitable for mobile battery-powered solutions. [11]

- *Texas Instruments TMP112* is a high-accuracy, low-power, digital temperature sensor. It is designed for various applications from portable and battery-powered solutions to general temperature merasurements in industrial controls. [40]

- *Secure Digital (SD)* memory card is specially designed to meet the security, capacity and performance requirements in newly emerging consumer electronic devices. The standard capacity of SD memory card is up to 2 GB. However, High Capacity SD (SDHC) extends the maximum capacity to 32 GB and Extended Capacity SD (SDXC) up to 2 TB. [35]. In this study, the maximum capacity of SDHC card shall be considered as the maximum available memory capacity. This restriction shall be taken into considerations while designing the sensor software. Moreover, the memory card shall be capable of recording one estrus cycle at minimum (approximately 21 days).

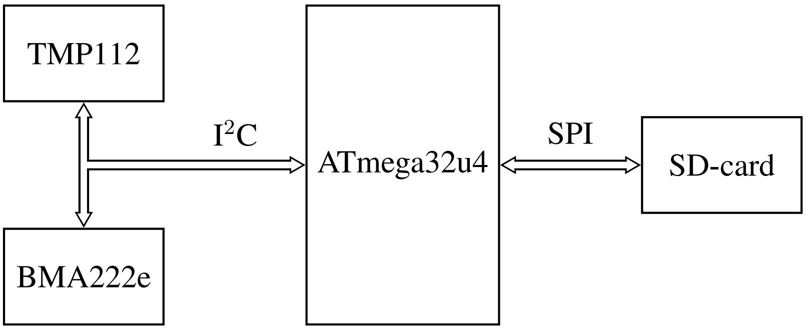

Figure 4: The principal block diagram of the sensor device. The temperature sensor and accelerometer are connected to the micro-controller via the same I^{2}C-bus, whereas, SD-card is connected via SPI-bus. Power connections nor USB are not represented in this figure.

In addition to the hardware components, the hardware configurations utilizes two serial interfaces, *inter-inegrated circuit (I^{2}C) bus* and *serial peripheral interface (SPI)*. They are explained briefly in the following.

- *Inter-Inegrated Circuit (I^{2}C) bus* which is some times referred as *Two-Wire Interface (TWI)* is a serial communication interface developed by Philips Semiconductor. The first version of the I^{2}C was released in 1982. The design of it is rather simple hence, it requires only only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), with pull-up resistors [41].

- *Serial peripheral interface (SPI)* [37] is a more complex serial interface than the (I^{2}C) bus. It requires at minimum of three parallel wires in three-wire mode [11] *signal select* ($\bar{SS}$), *serial clock* (SCK) and *serial data input/output* (SDI). However, in normal four-wire mode the data input and output are in separate lines, *master out, slave in* (MOSI) and *master in, slave out* (MISO).

As we have now introduced the main components of the sensor device, the communication principles of the device are represented in figure 4. As shown in the picture, SD memory card utilizes SPI serial interface whereas temperature sensor and accelerometer are connected into same I^{2}C bus. Naturally, all of these hardware components are assigned to their specific tasks with respect to their functional description. Therefore, their essential functionalities are discussed in the following topics.

Micro-controller ok

The *micro-controller unit (MCU)* is the core component of the sensors device in this study. That is, the micro-controller is responsible of execution of the software flow whenever the device is powered. Typically, the first executable task in the software flow is the system initialization. Normally, system is initialized once in software flow and each time after resetting or starting the micro-controller. During the initialization, the micro-controller sets up its control register according to the configurations in its program memory. These registers contains the configurations for the *general purpose input output* (GPIO) pin configurations as well as timers, serial interfaces and other featured functionalities. In addition to self configuration, the micro-controller initializes connected hardware via serial interfaces according to the configurations in the software memory. Next, after initialization the micro-controller begins the execution of the actual software flow. Typically, this flow is a repetitive software loop including variable tasks and events. The hardware initialization and configurations as well as the application tasks and events are discussed more the software section 3.1.2.

As discussed earlier, Atmel ATmega32u4 [10] is the micro-controller unit of the sensor device in this study. It is a high performance, low power micro-controller for various applications. Therefore, we will discuss of its features and functionalities more in detail. In spite of the wide range of its features, only the most fundamental properties are covered in the following discussion.

- $32\,\mathrm{kB}$ *of In-System Self-Programmable Flash memory* is the memory space for the actual program storage. Furthermore, the memory space is divided into two sections, Boot program Section and application Program section.

- *USB 2.0 Full-Speed/Low-speed Device Module* provides interface to write on the in-system self-programmable flash memory of the controller. Thus, it enables uploading the application software without external USB module. Additionally, the interface contributes serial communication between computer and the device.

- *General I/O* consists of 26 programmable I/O lines. These lines can be set as input or output separately. Furthermore, specific input pins can be configured as interrupt pins. Typically, interrupts are utilized for triggering events in the software flow. Additionally, interrupts are also a required feature in serial communication events.

- *Interrupts* are occasions that usually set an interrupt flag. Next, the *interrupt service routine (ISR)* corresponding the interrupt flag is executed with respect to its priority. Optionally, it is possible to execute a task directly in the interrupt. However, it temporarily disables other interrupts, hence, it is not recommended. Furthermore, there are internal and external interrupts. The source of internal interrupts are inside the micro-controller itself, as timers. Correspondingly, the source of external interrupts are external devices such as sensors or memory storages. Additionally, the interrupt can be enabled individually. The main advantage of utilizing interrupts is to execute tasks on-demand instead of periodically.

- *Watchdog Timer (WDT)* of ATmega32U4 employs a separate on-chip $128\,\mathrm{kHz}$ oscillator for creating timer events. If the WDT is enabled it is capable of resetting the entire micro-controller unless, the WDT is not reset regularly. Normally, WDT is utilized for recovering from deadlocks, unintentional conditions without exit. However, it is possible to utilize the watchdog timer as a wake-up timer in low power modes.

- *Power Management and Sleep Modes* allows the user to tailor the power consumption according the application requirements. Power management is beneficial feature specially with battery-powered solutions when regular re-charging is not effortless to accomplish.

- *SPI and I^2C* serial bus interfaces are used for device-internal communication of the micro-controller and the sensors as well as the SD memory card. The serial interfaces were briefly discussed in section 3.1.1.

Accelerometer ok

As discussed previously, a micro-controller is the core component of the hardware design of the sensor device. Similarly, an accelerometer is the essential sensors providing data for data recording as well as algorithm development. In this study, the utilized accelerometer is the Bosch Sensortech BMA222E [11]. It is a triple-axial accelerometer used for measuring the change in the motion. Additionally, it can be employed in position recognition in constant situations. The axises of the accelerometer are perpendicular. Therefore, it is capable of measuring in all direction. Additionally, it contains several advantageous on-chip functions. It is capable of prior and post filtering acceleration measurements. Additionally, it can detect various conditions and trigger interrupts in result. Moreover, the filters as well as the detectable conditions are highly configurable. These configurations are discussed in the following description. Nevertheless, the accelerometer contains several functionalities without use-cases in this study. Consequently, those features are not covered in the discussions. However, some of the features have no direct use case in this study but are applicable in the future studies that are considered in section 4.3. Additionally, the sensor configurations varies between two software implementations in this study and they are discussed in the sectin 3.1.2. The kay features of the BMA222E accelerometer are as follows.

- *On-chip interrupt controller* is capable of generating interrupts from various conditions. The conditions are configurable with respect to time or numerous motion statuses. On-chip interrupt controller yields an opportunity to create device applications even without a micro-controller. However, in this study all the interrupts are processed in the micro-controller. Nevertheless, the interrupt controller is utilized to create favorable sampling events instead of continuous sample recording.

- *On-Chip FIFO Register* is capable of storing up to 32 data frames. Depending on the configurations, one frame contains measurements from chosen or all three axis. Additionally, it is configurable whether the data in the register is filtered or unfiltered. Furthermore, the register contains information whether the data is new or old data.

- *Range* of the acceleration measurements is adjustable within four preset ranges, $\pm 2\,\mathrm{g}$, $\pm 4\,\mathrm{g}$, $\pm 8\,\mathrm{g}$ and $\pm 16\,\mathrm{g}$. However, increasing the range decreases the resolution and vice versa. Thus, selecting of appropriate range shall be considered.

- *8-bit Resolution* is applicable for both, acceleration and temperature measurements. As discussed, the range of acceleration measurements is adjustable and affects the resolution. Consequently, the resolution is from $15.63\,\mathrm{mg}$ per *least significant bit (LSB)* to $125\,\mathrm{mg}$ per LSB. The Nevertheless, the temperature resolution is fixed to $0.5\,^{\circ}\mathrm{C}$ per LSB.

- *Low Pass Filter* enables removing of high frequency distortion from the measured signals. Thus, no additional low pass filtering is needed in the micro-controller application. The low pass filter of the accelerometer is configurable with preset frequencies from 7.81 Hz to 1000 Hz. The bandwidth configuration effects the interval, how frequently, new data frame is readable from the register of the sensor. Roughly, the values are update twice often than the filter bandwidth according the Shannon-Nyquist sampling theorem.

- *Offset Compensation* allows removing offsets from the measured signals. At sea level, there is always approximately $1\,\mathrm{g}$ offset present. Specially in integrative calculations the presence of offset accumulates and yield misleading results.

- *SPI and I^{2}C digital interfaces* are necessary interfaces for configuring the sensor as well as data transmission from the accelerometer from the micro-controller. It is configurable, which interface is utilized.

- *Low Power Consumption* is beneficial in portable and battery-powered solutions such as the sensor device in this study. Additionally, the accelerometer enables improvement of the power consumption with several power modes. However, the power modes are not applicable in this study because, one of the core ideas is to utilize the accelerometer as effectively as possible instead of more power consuming micro-controller.

- *On-chip Temperature sensor* of the controller provides a resolution of $0.5\,°\mathrm{C}$. The accuracy of the BMA222E is relatively low in comparison to the TMP112 temperature sensor discussed later in this section. Nevertheless, it is useful as a reference or verification value with the measurements of the TMP112 temperature sensor. [11]

Temperature Sensor ok

Whereas the accelerometer is the main sensor of the sensor device, a temperature sensor is considered as a supportive sensor. Furthermore, the accelerometer contains a built-in temperature sensor. However, its resolution is relatively low. In contrast, Texas Instruments TMP112 is a high-accuracy temperature sensor for various applications from portable devices to industrial controls. It is specially designed for replacing *negative temperature coefficient (NTC)* and *positive temperature coefficient (PTC)* thermistors in high accuracy applications. In this study, the temperature sensor is used for monitoring the skin temperature of the cow. The target of the monitoring is to find correlation between the change of skin temperature and ongoing estrus or pro-estrus. Therefore, in the case of positive correlation the temperature sensor can confirm the detected estrus. In contrast, a negative correlation can confuse the results and yield a need for further studies. The properties of the TMP112 temperature sensor are described in the following.

- *High accuracy* of
  - $0.5\,°\mathrm{C}$ in range from $0\,°\mathrm{C}$ to $65\,°\mathrm{C}$
  - $1.0\,°\mathrm{C}$ in range from $-40\,°\mathrm{C}$ to $125\,°\mathrm{C}$

without calibration. Furthermore, instructions for the calibration of the sensor are provided in the data sheet.

- *High resolution* of $0.0625\,°C$ in both, 12-bit and 13-bit modes
- *Low power consumption* in two different power modes:
  - $10\,\mu A$ in active mode
  - $1\,\mu A$ in shutdown mode
- SMBus^{TM}, Tow-Wire and I^2C digital interfaces
- supply voltage range from $1.4\,V$ to $3.6\,V$
- *Conversion rate* from $0.5\,Hz$ to $8\,Hz$
- *12-bit resolution* from $-55\,°C$ to $128\,°C$. The sensor has an 13-bit mode, when the measu rement range is up to $150\,°C$. [40]

In this study, the temperature sensor is soldered directly on the circuit board. In result, the sensor is not in direct skin contact with the cow. Therefore, the skin temperature is conducted from skin to the sensor via heat conducting aluminum tape.

3.1.2 Software

Previously, we discussed of the data recording hardware and the primary components of it. As stated earlier, the sensor device is a prototype of a dairy cow sensor device. Furthermore, it was customized for data recording by adding a SD card slot on its serial peripheral interface (SPI) bus. Otherwise, no additional modification was required. In contrast to the already existing hardware design, there software had to be implemented from a scratch. Therefore, in this section we discus of the software design as well as the actual software implementation. Additionally, we introduce the software tools form implementing the micro-controller application software for the sensor device. The software design required prior testing of the hardware capabilities as well as studying of the available software libraries. Conversely, the software implementation was more straight forward process based on the software design.

In this study, the software tool for sensor application implementation was Arduino Integrated Development Environment (Arduino IDE) [6]. The Arduino IDE provides a fast and easy environment for implementing micro-controller applications without prior expert-level knowledge on micro-controllers. Additionally, Arduino and Arduino community provides comprehensive set of software libraries for various micro-controller applications. The following list describes the fundamental software libraries in this study:

- *Wire* library provides functionality for Inter-Integrated Circuit (I2C) communication. "This library allows you to communicate with I2C / TWI devices. On the Arduino boards with the R3 layout (1.0 pinout), the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. The Arduino Due has two I2C / TWI interfaces SDA1 and SCL1 are near to the AREF pin and the additional one is on pins 20 and 21." [7]

- *SPI* "This library allows you to communicate with SPI devices, with the Arduino as the master device." [8]

- *SD* "The SD library allows for reading from and writing to SD cards, e.g. on the Arduino Ethernet Shield. It is built on sdfatlib by William Greiman. The library supports FAT16 and FAT32 file systems on standard SD cards and SDHC cards. It uses short 8.3 names for files. The file names passed to the SD library functions can include paths separated by forward-slashes, /, e.g. "directory/filename.txt". Because the working directory is always the root of the SD card, a name refers to the same file whether or not it includes a leading slash (e.g. "/file.txt" is equivalent to "file.txt"). As of version 1.0, the library supports opening multiple files." [9]

- *EnableInterrupt* "enableInterrupt- Enables interrupt on a selected Arduino pin. disableInterrupt - Disables interrupt on the selected Arduino pin." [34]

- *RingBuf* "This is a simple ring (FIFO) buffer library for the Arduino. It is written in vanilla C, and can easily be modified to work with other platforms. It can buffer any fixed size object (ints, floats, structs, etc...)." [44]

However, the some advantageous features of micro-controllers are not included in standard Arduino libraries. Thus, profound familiarization with the micro-controller data sheet is beneficial for achieving the highest level of performance. In addition to the standard libraries, Arduino community provides plenty of open-source software libraries for various applications.

Furthermore, the Arduino programming language is merely a set of C/C++ functions. Thus, implementing

Additionally, the driver interfaces provided by the sensor manufacturers were not fully compatible with the Arduino. Thus, both driver interfaces were included in the implementation of the device software.

Data logs are recorded on a regular .txt-file on the SD card.

the actual implementation is created in Arduino Integrated Development Environment (Arduino IDE) [6]. The software implementation includes the conduction of the driver interfaces of both, accelerometer and temperature sensor.

This subsection discusses of the data recording software. In contrast to the hardware, the software improved during this study. Thus, the first software implementation differs significantly from the latter. As a matter of fact, the first software implementation based on previous researches and intuition, whereas the latter based on the results received from the first recorded set of data. That is to say, the results of the first set of data formed a general view on typical behavior of a cow, while the purpose of the following data sets were used for the actual estrus detection. *Tähän vielä jokin tarnsitiofraasi, ehkä?*

The software were implemented in Arduino IDE (*Integrated Development Environment*). The use of Arduino IDE offers effective environment for implementing embedded software without expert-level knowledge on micro-controllers. The Arduino provides extensive libraries... However, the driver interfaces for accelerometer and temperature sensors as well as the entire flow of the software were self-implemented.

First Software Implementation

The starting point for implementing the first software included only a cursory conception of the behavior of a diary cow. Therefore, the properties of the accelerometer described in section 3.1.1 are treated with care in order to avoid loss of relevant data. In conclusion, a high data rate was prioritized over other features. Furthermore, it was decided not to use offset compensation since, it could cause unawareness of the pose of the device.

The accelerometer sets a hard restriction of 2000 Hz for the maximum data rate. However, the usage of secure digital (SD) memory card as a data storage limits the data rate even more as discussed in section 4. That is, the duration of the SD file operations exceeds the disposable time at high data rate and, thus, causes loss of data. In contrast, a low data rate could cause a loss of possibly relevant features on higher frequencies. Therefore, the selection of the data rate is more or less a trade of between data losses and data bandwidth.

The flow of the software consists of tasks and conditions. Each task contains a single function or a sequence of functions the micro-controller must execute before proceeding to the next task or condition in the flow. In this flow, the conditions are used to decide whether a task is being executed or not. Alternatively, a condition may be followed by an entire branch of tasks instead a single task. The conditions of the software flow and their explanations are:

- *Interrupt* is true if an interrupt flag is set by an interrupt from the accelerometer which in this case is a certain FIFO buffer level.
- *Timer* condition is used for reading temperature data from both of the sensors periodically.
- *Timeout* is a backup feature if an interrupt is being missed and, therefore, no more interrupts received.
- *Errors* is true if any of the predefined errors have occurred during the execution of the main loop.
- *Buffer empty* condition checks the buffer level of the micro-controller. If the buffer level is non-zero the contents of the buffer will be written to the SD card.
- *Save data*

and following tasks:

- *Initialize* task (which is analogous for *setup* function in Arduino IDE) is executed only once right after the device is powered up. This task initializes the desired configurations for the accelerometer and the temperature sensor. It
- *Read acceleration data* task reads the data from the FIFO buffer of the accelerometer and stores the acceleration data into the FIFO buffer of the micro-controller.

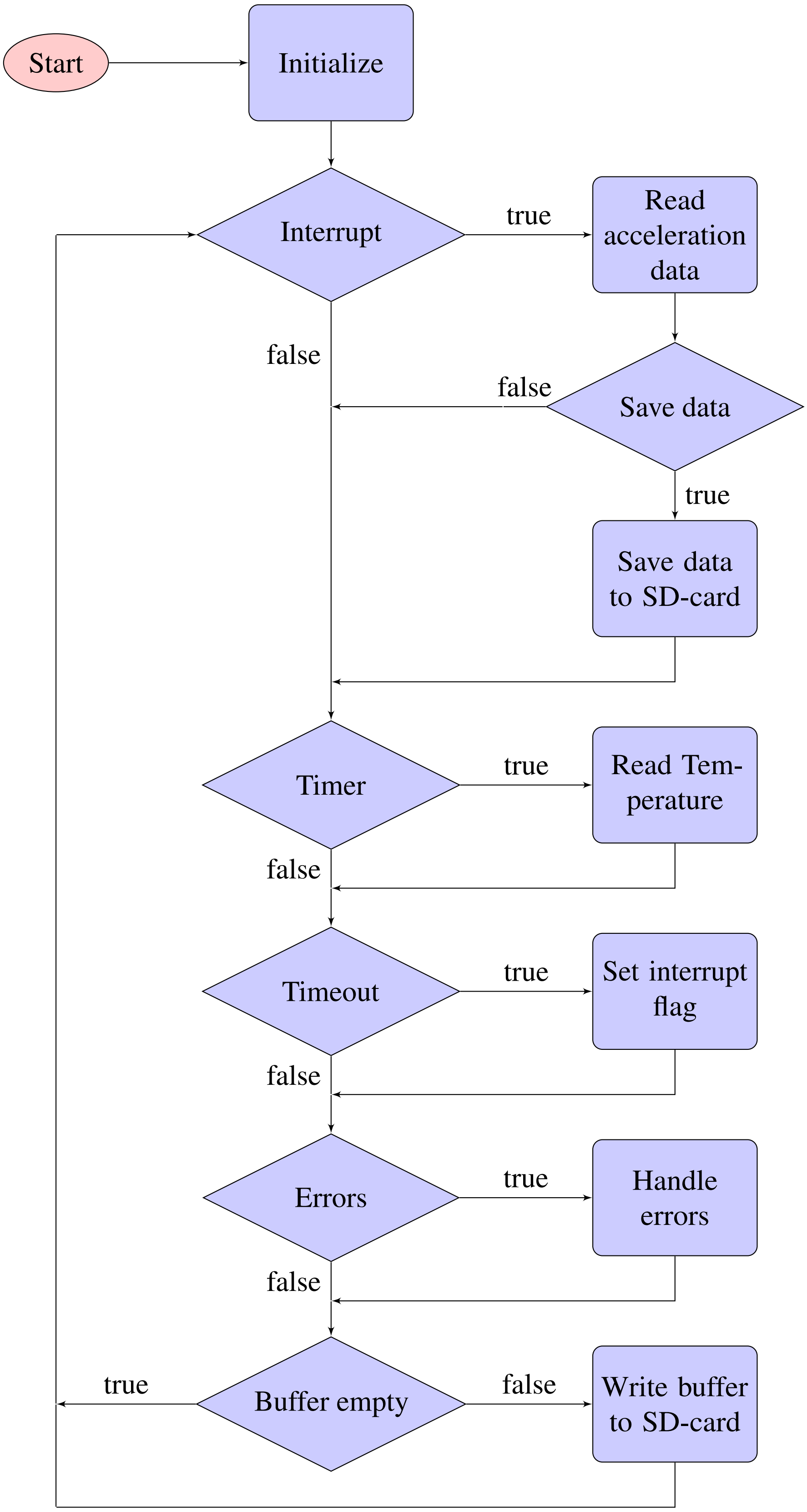

Figure 5: The flow of data logging program

- *Save Data to SD-card* task saves the data written on the SD card. That is, the file where data is being written must be closed in order to ensure the written data is being saved. Once the file is closed for saving the data it has to be re-opened for continuing the writing process. Alternatively, if the size of the current file exceeds a preset limit, a new file is opened. Furthermore, the duration of the open and close file operations could exceed the time required to fill up the FIFO buffer of the accelerometer. Thus, the FIFO buffer is being read empty before and after closing the current file and once more after opening a file.

- *Read temperatures* task reads the temperature data of both sensors and stores the value to the memory of the micro-controller. No temperature data are buffered. Thus, only the latest values are written to log file on SD card.

- *Set Interrupt flag* sets the interrupt flag without an interrupt if a timeout condition is met. Once the interrupt flag is set, the micro-controller will read the acceleration data from the FIFO buffer of the accelerometer.

- *Handle errors* task handles predefined errors if one or more of them have occurred. In practice, this task writes the name of the occurred to the text file and resets the error flag.

- *Write buffer to SD-card* writes a single line of acceleration data from the FIFO buffer of the micro-controller into a text-file on the SD card.

Second Software Implementation

The approach for recording the second data set differs significantly from the first one. The first software implementation attempted to maximize the data rate, whereas the second focused on power saving.

The work flow consists of the following conditions:

- *FIFO-level > 0* condition is true if the FIFO buffer level of the accelerometer is non-zero.

- *WDT flag* condition is true if the watchdog timer has set a watchdog timer (WDT) flag.

and the following tasks:

- *Initialize*

- *Read acceleration* task reads all the data from the FIFO buffer of the accelerometer and stores the data into a FIFO buffer of the micro-controller.

- *Read temperature* task reads the temperature of both of the sensors, accelerometer and temperature sensor.

- *Open file* task opens a file for writing the data. The task opens a new file if the size of the current file exceeds a preset limit for maximum file size.

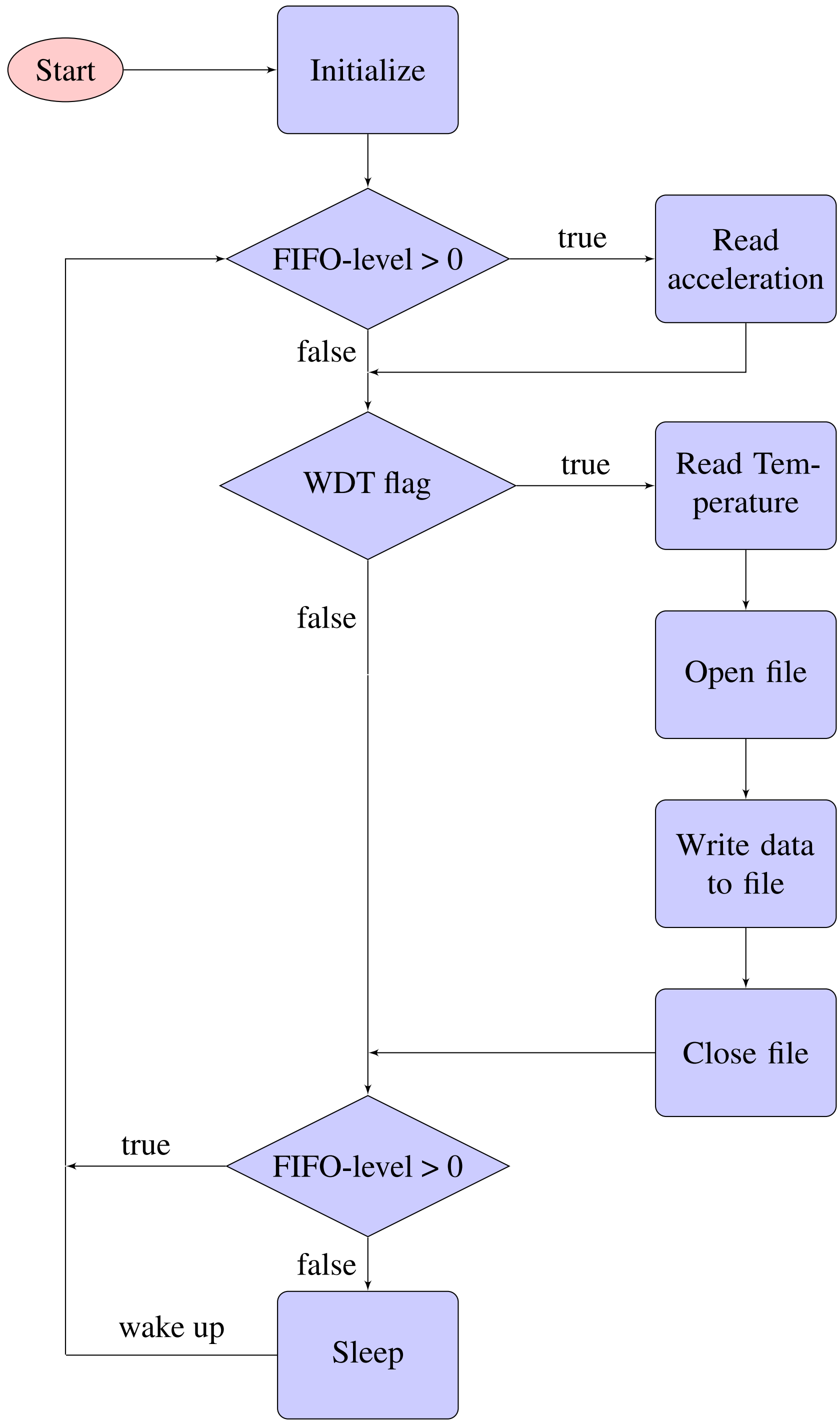

Figure 6: The program flow of the second data logging software

- *Write data to file* writes all the buffered acceleration data into the opened file on the SD card. In addition, the task writes the temperatures of both of the sensors, the number of occurred watchdog timer interrupts and the up-time of the micro-controller into the text file.
- *Close file* task closes the opened file in order to ensure the written data is being saved.
- *Sleep* task puts the micro-controller in a power saving mode. That is, all the other functionalities but watchdog timer and interrupts are disabled in order to minimize the power consumption. The micro-controller remains in the sleep until the watchdog timer or an interrupt from the accelerometer wakes up the micro-controller. After waking up, the

3.1.3 Procedure

All the data used in this study were recorded on a dairy farm in coordinates $\mathrm{N}63°6'3.6''$ $\mathrm{E}23°10'35.5''$. The breed of the farm consists of Ayrshire and Holstein cows, which were also used for data recording.

The data was recorded in two separate occasions. Furthermore, the software implementation for these occasions differed significantly as discussed in sections 3.1.2. The sets of data were recorded in a farm at The breeds on the farm are …

First Data Set

The primary objective for the first recorded data set was achieving an overview on the behavior of a dairy cow. Thus, the sensor device was attached to the neck of a cow with a trail camera. In the beginning of the recording of the first data set, only one sensor device and one trail camera were available. Therefore, only one cow could have been chosen for recording simultaneously. In spite of the primary objective of observing the behavior, it was desired to obtain estrus data as well. Thus, the cattle tender assisted choosing a cow that was estimated having an estrus during the recording process. The maximum recording period was approximated up to 16 days, hence the capacity of the SD card was $8\,\mathrm{GB}$ and the recording required approximately $500\,\mathrm{MB}$ per a day.

The recording of the first data set was started on Friday September the 9$^{\text{th}}$ at the farm. The sensor device was attached to the neck of a selected cow with a trail camera. The recording lasted till Sunday September the 18$^{\text{th}}$. After the sensor and the camera were detached from the neck of the cow, the recorded data from both, the sensor device and the trail camera were copied on a computer for analysis.

The trail camera as well as the sensor were without proper time stamp. Thus, they were out of synchronization. Consequently, making of conclusion between the recorded data and video was not reasonable. However, it seemed that there were recognizable patterns in the behavior of the cow e.g., eating of feed and walking. Because of these promising results, the same approach was attempted with the further data recordings. We shall discuss of it later in this study.

Second Data Set

The second data set was recorded in two phases. First phase was recorded from December the 14th to 15th. The second phase was recorded instantly after the first and lasted until January the 20th. The cows selected for the data recording were such that they were estimated to have an estrus during this period.

MAINITSE että lehmillä oli heatime!!!

Figure 7: Cow wearing a sensor device. The axis directions are illustrated as red arrows in the figure. The sensor is in parallel with a commercial Heatime device shown in this figure.

ARvioidaan toisen (ja kolmannen) datasetin tuloksia, mutta myös laitteistoa ja ohjelmistoa. Jotain johtopäätöksiä.

3.2 Data Processing

This subsection discusses about the methods used in data analysis. The used methods consists of pure visual observations as well as statistical analysis and more advanced digital signal processing. However, the visual observations remain the primary tool before and after applying any statistical or signal processing method.

Parsing the data

The sensor data was recorded on .txt-files on the SD card. After recording the data long enough, the sensor was removed from the cow and the log files were copied on a computer.

Put here a sample of log files?

3.2.1 Statistics

In this study, statistical methods are included in the estrus detection algorithms discussed in section 3.3. Furthermore, statistics are used in analysis of duration of SD file operations

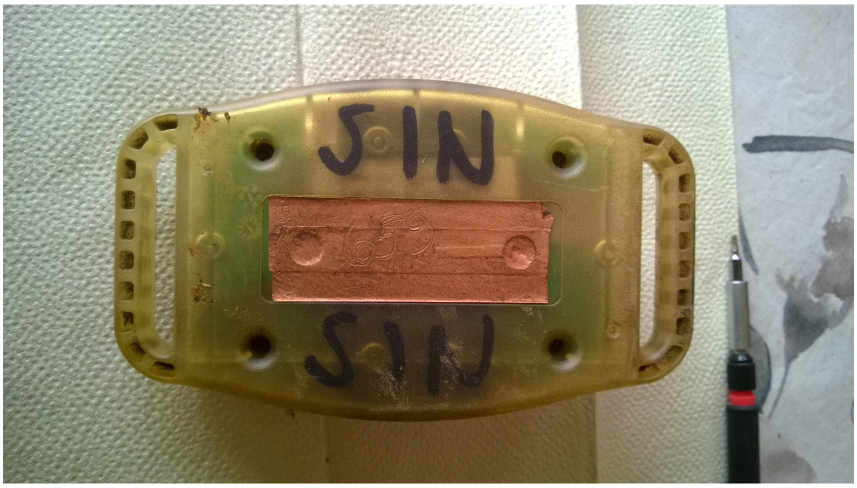

Figure 8: The heat conducting tape on the bottom-side of the sensor device. The purpose of the tape is conducting the skin temperature of the cow to the temperature sensor on the circuit board.

in section 4.1.

The statistical methods used in this study are defined as follows [36]:

- *Mean* is used for describing the most common value of the data set. It is defined as the sum of all values divided the number of values:

$$\bar{x} = \frac{\sum_{i=1}^{n} x(i)}{n} \tag{1}$$

- *Median* is another number for describing the most common value of the data. In contrast to the mean, it is not that sensitive to exceptionally large or small values. It is defined as the middlemost value of sorted data. If the number of values in the data is even, the median is the sum of two middlemost values divided by two.

  Median is the middlemost number of the data set sorted from smallest to largest numbers. If the number of elements in array is even, the median is the mean value of the two middlemost values.

- *Variance* describes the expectation of the squared deviation of a random variable from its mean, and it informally measures how far a set of (random) numbers are spread out from their mean:

$$s^2 = \frac{\sum_{i=1}^{n} (x(i) - \bar{x})^2}{n} \tag{2}$$

- *Standard deviation* describes the amount of variation in the data set. A low standard deviation means the values tend to be close to the mean value, whereas a high value means the values tend to be far from the mean. In contrast to the variance, the standard deviation describes the “typical” distance between the values and the mean. The standard deviation is actually the square root of variance:

$$s = \sqrt{\frac{\sum\limits_{i=1}^{n}(x(i) - \bar{x})^2}{n}} \tag{3}$$

- *Minimum value* is the smallest value in the data set.
- *Maximum value* is the largest value in the data set.
- *Range* is the difference of the minimum and maximum values.

3.2.2 Fourier Transform

Fourier transform is linear mapping from time domain to frequency domain.

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x)^{-i\omega x} \mathrm{d}x \tag{4}$$

$$F_n = \sum_{k=0}^{N-1} f_k e^{-i(2\pi k/N)n} \quad , \mathrm{n} = 0, \ldots, \mathrm{N} - 1 \tag{5}$$

Discrete Fourier transform (DFT) is linear mapping of a signal from time domain to frequency domain [33]. That is, a time variant sample can be transformed into frequency domain for providing the frequency spectrum of the data. In this study, the frequency spectrum of cow behavior is valuable information in deciding the the bandwidth of the accelerometer.

Frequency analysis is important in designing of sampling rate as well as filters. According to the Shannon-Nyquist theorem, the minimum sampling rate shall be at least double the highest frequency component. We will discuss of filters in the following subsection.

Normally, DFT would require approximately N^2 iterations for mapping the time domain to the frequency domain. With large samples this becomes mathematically demanding and time consuming process. Therefore, there are several approximation approaches called Fast Fourier Transforms (FFT). They require less computing and time. However, the results are not as exact as it is with DFT. In this study we will use the FFT functions included in the standard libraries of Matlab software. The theory behind FFT is quite deep and, therefore, it is not discussed in detail in this study. Nevertheless, we will introduce the use of the fft function of Maltab as follows. For the fft, it is required to define the length of the sample as well as the sampling frequency. $\mathbf{O}(\mathbf{N} \log \mathbf{N})$

Next we will create a vector with frequencies from zero to half of the sampling frequency. Actually, the fft of the Matlab does not straight consider the true frequencies. Therefore, the user himself need to set the frequencies correctly. The fft function results the frequency data in complex numbers. Therefore, it is required to get the absolute value of the results. Yet, the the results are two sided data. Thus, we need to exclude the other half of it. Lastly, we correct the amplitude and plot the resulting vector with the frequency vector we created earlier.

3.2.3 Digital Filters

In general, there are four different kinds of filters, high-pass, low-pass, band-pass and band-stop filters. The purpose of the filters is to remove distortions and undesired features from the signal. Usually, filters are applied at least in the beginning of the data processing sequence. However, filters may be applied in several phases depending on the applications. Low-pass filters are good in removing high-frequency noises and components above the research frequencies. Conversely, high-pass filters are used e.g., removing static off-set components in signal. We will discuss of the use cases of both of these components later in this study. Moreover, filters can be classified as analogous and digital filter. Typically, analog filters are configured in hardware and they are a combination of such analog electrical components as resistors, capacitors and inductors. The type and bandwidth of the filter depends on the design.

Digital filters require digital technology, which means discrete time signals and discrete time behavior of the circuit.

The Bosch Sensortec BMA222e accelerometer provides two on chip filters: one 2nd order low-pass filter and another 1st order high-pass filter for offset compensation [11]. Digital filters can be divided into two categories, filters with finite impulse response (FIR) and filters with infinite impulse response (IIR). The main difference between these filters is that the output of a FIR filter is dependent only on the input, whereas, the output of an IIR filter is dependent also on the previous outputs of the filter. Therefore, [14] [21].

The digital signal processing methods consists of applying filter and other mathematical methods. In these methods, any features of the sensors could be simulated afterwards instead of using the features of the sensor. E.g., offset compensation using the low pass filter causes the loss in information of orientation of the device.

Basically, the factors of the filter could be chosen freely. However, commonly the target total gain of the filter is 1.

Filters with infinite impulse response (IIR):

$$y[k] = b_0x[k] + b_1x[k-1] + b_2x[k-2] + \cdots + b_{n-1}x[k+1-n] + b_nx[k-n] \tag{6}$$

$$\begin{aligned} y[k] = b_0x[k] + b_1x[k-1] + b_2x[k-2] + \cdots + b_nx[k-n] \\ -a_1y[n-1] - a_2y[n-2] - \cdots - a_my[k-m] \end{aligned} \tag{7}$$

and filter with finite impulse respones (FIR):

3.2.4 Sliding Window

In this study, windowing is means analysis and calculus of data in segments instead of entire data set. The windowing method in this study is analogous to windowing in signal processing [38, 29] ...

The integration period = window

Windowed mean value:

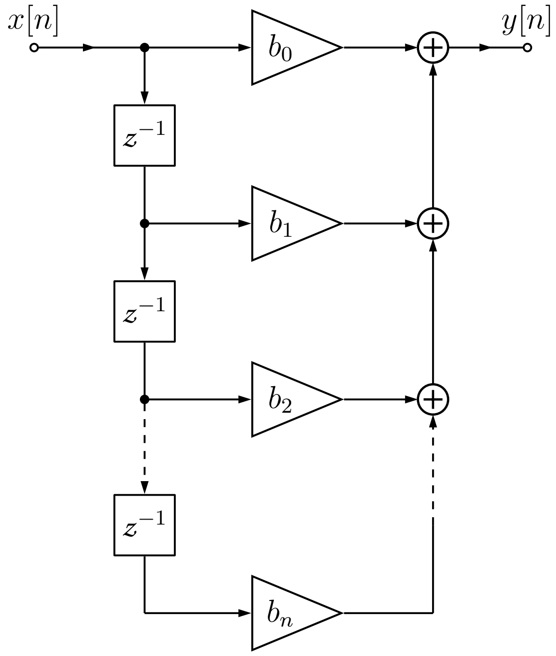

Figure 9: An example of an n^{th} order FIR filter

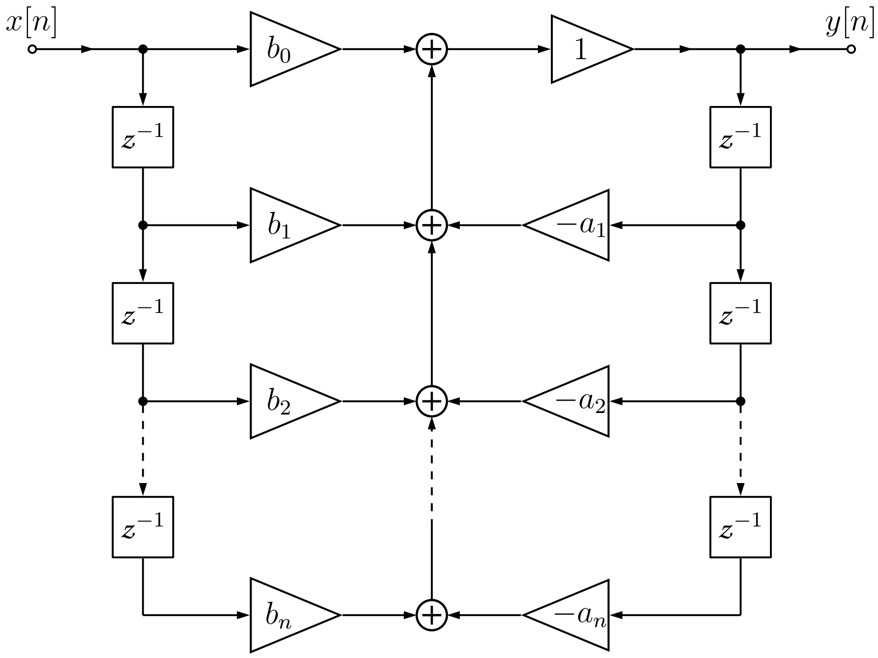

Figure 10: An example of an n-th order IIR filter in direct form I

$$\bar{x}(k) = \frac{\sum_{i=k-n}^{k} x(i)}{n}, \tag{8}$$

where n is the size of the window.

3.3 Estrus Detection Algorithms

In previous subsection we surveyed through the micro-controller software implementation of the sensor device. Additionally, we discussed of recording the data to log files. Consequently, this subsection focuses on both, parsing of the log files and specially the estrus detection algorithms. In this study, there are three different algorithms for estrus detection. Nevertheless, they all attempt to detect rather the proestrus than the actual estrus. This differs from the traditional estrus detection by the cattle tender we discussed in section 2.1.2. Traditionally, a cow allowing of being mounted by other cows is reliably considered to be in estrus (standing heat). However, detection of being mounted with accelerometer based sensor is rather unreasonable approach. Fortunately, the activity level of the cow increases in proestrus. Thus, all the following estrus detection algorithms attempt to detect rather the proestrus than the actual estrus. In despite of this similarity, all the following algorithms have their own approach to the topic. The algorithms are discussed more in detail in the following subsections.

As stated earlier, the sensor device software were implemented in Arduino IDE, an integrated development environment for micro-controller applications. However, the scope of this study is not in developing the estrus detection algorithms for the device directly. Consequently, it eases of implementing the sensor software. Additionally, it enables to choose freely more applicable software tools for developing and testing algorithms with recorded data. Nevertheless, one goal of this study is to create algorithms suitable for micro-controller platforms. The software tool for algorihtm development in this study is *MathWorks MATLAB* (later only Matlab) [28]. Matlab is a desktop environment created for various iterative analysis and design processes. It uses a programming language that expresses matrix and array mathematics directly. Furthermore, in contains numerous applications for various needs of engineers. In this study, the the interest is in data plotting and filter designing. Moreover, use of existing Matlab functions altogether with self-implemented functions and scripts provides advisable environment for algorithm development.

Considering the activity measurement in wearable devices for people, the activity is typically measured as sum of absolute values of all the axis. In studies, it has been proved it to match to the real energy consumption better than the absolute value.

The main features of all the following algorithms are as follows:

1. **Process data** — Data processing consists of algorithm specific data functions. These functions may include data filtering and other computations. These functions are discussed in detail in the following subsections. All the remaining phases of the algorithms follows the same pattern discussed below.

2. **Sum results** — The results of data processing are summed within time windows. The length of the time window should approximate the duration of the proestrus. Additionally, these time windows shall overlap in order to provide more continuous impression of ongoing state of estrus cycle. Furthermore, excessively long time windows without overlapping could delay the detection of the estrus. Thus, cause the failure of insemination as a consequence.

$$u(k) = \sum_{i=nk-m}^{nk} x(i) \quad \text{, where} \tag{9}$$

k is the index of summed results, n is an incremental step size and m is the size of the time window.

3. **Remove offset** — The resulting data after summing may differ significantly within algorithm depending on the parameters as well as between the algorithms. Therefore, any existing offset in the data should be removed. This means adjusting the data so that the normal behavior appears around the zero. The median of the data describes the amount of the offset more reliable than mean value. Hence, median is less sensitive to proestrus peaks in the data.

$$u(k) = x(k) - \tilde{x} \quad \text{, where} \tag{10}$$

$\tilde{x}$ is the median value of x.

4. **Normalize** — After removing the offset, the data shall be normalized. In this case, the normalizing means scaling the data so that no extreme value shall exceed the range of from -1 to 1.

$$u(k) = \frac{x(k)}{\max |x|} \tag{11}$$

5. **Threshold** — Finally, thresholds are used for indicating the beginning and the end of the proestrus. That is, exceeding the first threshold indicates the beginning of the proestrus and next, the going below indicates the end of the proestrus. The thresholds for deciding whether the cow is in estrus or not should be separate. The end of proestrus signals to the cattle tender to prepare for insemination. The principle of the proestrus detection is presented in the following pseudo code.

6. **Plot data** — Plotting of the resulting data visualizes the results. Nevertheless, it does not effect to the results directly. However, plotting makes the algorithm evaluation quicker and easier by brief visual inspections. Furthermore, plotting can reveal some features that might not be detected in pure numeral form.

3.3.1 Activity Monitoring

The concept of the first estrus detection algorithm is analogous to an accelerometer based estimation of total energy consumption [20]. However, finding of any correlation between the activity and the energy consumption is not a focus of this study. Nevertheless, this algorithm utilizes the same method for determination of the cow activity level.

The first algorithm is based on pure analysis of full the data set. The algorithm is analogous to an estimation of energy consumption using accelerometers ...

1. **Filter** — The data consists of an offset of approximately $1\,\mathrm{g}$ which effects to results. Actually, cows are rather passive animals and existing offset overrules the actual motion data. Therefore, removing the offset improves the performance of the algorithm and it shall be the first step in the algorithm sequence.

$$y(k) = a_0x(k) + a_1x(k-1) + a_2x(k-2) - b_1y(k-1) - b_2y(k-2) \tag{12}$$

2. **Compute** — The actual core of this algorithm is the length of the acceleration vector and it is defined as
$$u(i) = \sqrt{x^2(i) + y^2(i) + z^2(i)} \tag{13}$$

3. **Sum results** — With this algorithm there is no need for summing, hence, the algorithm uses the length of the total acceleration vector instead of calculating vectors separately.

4. **Remove offset** — The median value of the resulting array is considered as the mean offset. The offset is deducted from all the data points.

5. **Normalize** — After removing offset, remaining data is divided by the maximum of the absolute value of the data. In result, all the data points are between ± 1.

6. **Threshold** — Next, exceeding of preset threshold is considered as the beginning of the proestrus. Conversely, falling below second threshold indicates the end of proestrus and the beginning of estrus.

7. **Plot data** — Finally, the data is presented in plot. The plot shows the algorithm results against time. In addition to the data, also the beginning and the end of proestrus are shown in the plot.

3.3.2 Variance Detection

The first algorithm had a basis in continuous computation of continuous data stream. In contrast to the first algorithm, the second algorithm attempts to reduce the amount of required data. That is, using standard data samples in regular intervals instead of continuous stream. Furthermore, the calculus of the algorithms are different. The ground of the first algorithm was in summing of the lengths of the total acceleration vectors, whereas this algorithms attempts to determinate the level of variance

The first algorithm calculated the length of the total acceleration vectors. Thus, estimated the energy consumption. Alternatively, this second algorithm detects the level of variation in the acceleration data rather than the total amount of movement.

Originally, the concept of a variance based algorithm arose along with the analysis of the recorded data sets.

Originally, the concept of a variance based algorithm arose among the analysis of the first recorded data set. The data consisted of two distinct period. They differed in variance as well as in offset. In theory, both of these differences could have been used as a basis of an algorithm. However, the reliability of the offset is questionable, hence the pose of the device is not strictly fixed in the neck of a cow. Additionally, offset based algorithm would prevent the use of the offset compensation of the accelerometer. Consequently, it would set unnecessary restrictions for future development. Moreover, the difference in offset was mostly significant in the x-axis data. Conversely, the amount of variance appeared evenly on each axis. Furthermore, variance is offset independent. Thus, variance based algorithm does not require any high-pass filtering before data processing.

1. **Get a sample** — In this algorithm, the data is processed in samples instead of data stream. The length of a sample is variable. Furthermore, there are no restrictions for overlapping. However, the purpose of this algorithm is to reduce the amount of required data frames.

2. **Compute Variance** — A sample consists of predefined amount of data frames. Computing of variance means calculating the variance of single data sample. However, in this study we have all the data available. Thus, the sampling and computing the variance are combined in the following equation:

$$u_x(k) = \frac{\sum\limits_{i=nk-m}^{nk} (x(i) - \bar{x})^2}{m} \quad , \tag{14}$$

   where m is the size of the data sample, n is the distance between the samples and k is the index of the output data. The product nk shall not exceed the index of the maximum samples.

3. **Sum results** — The variance of each axis are calculated separately. However, in this study we will not compare the algorithms for each axis separately. Thus, the resulting variances are summed index by index as follows:

$$u_{tot} = u_x + u_y + u_z \tag{15}$$

4. **Remove offset** — Similarly as before, the median value of the resulting data is considered as the offset and it will be deducted from all of the data points.

5. **Normalize** — After removing the offset, the resulting data set is divided by the maximum of the absolute extreme value. In result, all the data points are between values ± 1.

6. **Threshold** — Next, exceeding of preset threshold is considered as the beginning of the proestrus. Conversely, falling below second threshold indicates the end of proestrus and the beginning of estrus.

7. **Plot data** — Finally, the data is presented in plot. The plot shows the algorithm results against time. In addition to the data, also the beginning and the end of proestrus are shown in the plot.

3.3.3 Inactivity Detection

The first two algorithms were based on relatively demanding computations considering micro-controller (MCU) environments. The first algorithm required a continuous data stream and calculating powers of two and square roots. The second algorithm did not require a continuous data stream. However, the algorithm included computing of variance which includes sum, square root and power of two as discussed in section 3.2.1. Furthermore, the available dynamic memory of micro-controllers are restricted. Therefore, it limits the maximum number of retained data points for computing the variance. In contrast, the third algorithm attempts to reduce the computation in the MCU. Therefore, it requires more advanced deployment of the features of the accelerometer. Accordingly, an interrupt driven approach becomes sensible. That is, the the MCU only counts the number of interrupt events whereas, the accelerometer performs all other computations.

In addition to transferring most of the calculus from MCU to accelerometer, the logic itself could be inverted. That is, monitoring inactivity instead of activity. For this perspective, the accelerometer provides the feature of *no motion detection* which was discussed in section 3.1.1. This feature suits fairly well to the aspect of transferring the computation from MCU to micro-controller and observing inactivity instead of activity. However, this study bases on recorded data instead of testing these configurations real animals. Therefore, the algorithm described below is purely a simulation of the features of the accelerometer. In consequence, the results achieved in this study might differ from those of real life.

This inactivity detection algorithm complies with the algorithm structure discussed earlier in this section. The data processing phase of this algorithm is as follows.

1. **Get Slope** — The slope is the acceleration difference. That is, the previous value subtracted from the current value:

$$u[i] = x[i] - x[i-1] \tag{16}$$

   Furthermore, the slope is offset independent. Hence, the slope is actually first order FIR filter with infinite attenuation at zero frequency, sampling frequency and its harmonics.

2. **Detect inactivity** — According to the specifications of the accelerometer, a no-motion interrupt is triggered if the absolute value of the slope remains below of a preset threshold for a preset duration of time. The following pseudo code represents the software implementation for the no-motion detection.

```
for (i = 0 ; i < length(x); i ++)
 if (x(i) > threshold)
  prev_i = i;
 if (i - prev_i > passivity_period)
  passivity(i) = 1;
  prev_i = 1;
```

In this study, the passivity array has the same length as the acceleration data array and it is initialized as zeros. During the execution, the algorithm processes through the entire data set. Meanwhile, any occurred no-motion condition yields the value of one into the passivity array. In consequence, the resulting passivity array consists mostly of zeros and few of ones within. Furthermore, the indexes of the ones are directly related to the time of the occurrence. Thus, it enables concluding the level of inactivity in certain range of time.

3. **Sum results** — sdf

4. **Remove offset** — The median value of the resulting array is considered as the mean offset. The offset is deducted from all the data points.

5. **Normalize** — After removing offset, remaining data is divided by the maximum of the absolute value of the data. In result, all the data points are between ± 1.

6. **Threshold** — Next, exceeding of preset threshold is considered as the beginning of the proestrus. Conversely, falling below second threshold indicates the end of proestrus and the beginning of estrus.

7. **Plot data** — Finally, the data is presented in plot. The plot shows the algorithm results against time. In addition to the data, also the beginning and the end of proestrus are shown in the plot.

Next, after these two specific phases of data processing phases the algorithm proceeds its normal sequence as discussed earlier in this section.

4 Results

In this section we will represent the results of this study. The results are achieved following the instructions of the section 3. First, we will discuss of the results in general. The discussion covers several topics and phenomena arisen during this study. The second subsection focuses on revealing the results of the estrus detection. Lastly in this section, we will discuss of conclusions based on the results. The conclusions includes a summary of algorithm evaluation, discussion of failures during this study and suggestions for future studies.

4.1 General Results

This subsection discusses about rather general results whereas the scope of subsection 4.2 is in the detection of estrus. In despite of the general nature of these results, they are advantageous considering research in future. Furthermore, some early stage results such as the frequency analysis discussed later in this section were used for improving the sensor device software. This subsection covers the frequency analysis of the first recorder data set as well as statistical analysis of the acceleration data and some interesting features of SD file operations.

4.1.1 Frequency Spectrum

The frequency spectrum of the first recorded data was analyzed using Fast Fourier Transform (FFT) as discussed in section 3.2. The Fourier Transform was applied on the raw data without any other signal processing methods such as offset compensation. Therefore, the frequency spectrum of each axis consists of significant components at zero frequency and nearby. These components could have been reduced using such offset compensation as a high-pass filter discussed in section 3.1.1. However, applying offset compensation to the data does not provide any additional information in frequency domain. Thus, awareness of the offset component is enough considering the usage of the results. The results of the FFT are represented in figure 11.

4.1.2 SD File Operations

In the first data recording process, it was desired to record the data in as high bandwidth as possible. The maximum bandwidth of the accelerometer were $1000\,\mathrm{Hz}$. Consequently, the update time was $0.5\,\mathrm{ms}$. However, the duration of SD file operations restricts the maximum recording bandwidth significantly. Finally, in the first data recording process the bandwidth was selected to be $125\,\mathrm{Hz}$. Yet, the file operations exceeded the time ...

SD card file operation times were recorded after the actual data recording using the original hardware. The original software was improved with timer operations in order to enable the recording of the file operation times. The file operation times had an affect on the data recording process. The time consumed for opening a file increased among the file size. The minimum opening time was $7.26\,\mathrm{ms}$ and the maximum time for $1.37\,\mathrm{GB}$ file a was $1440\,\mathrm{ms}$. The average value was $747.14\,\mathrm{ms}$. The file closing times varied from $8.04\,\mathrm{ms}$ to $160.03\,\mathrm{ms}$ and seemed not to be dependent on the file size. The average file

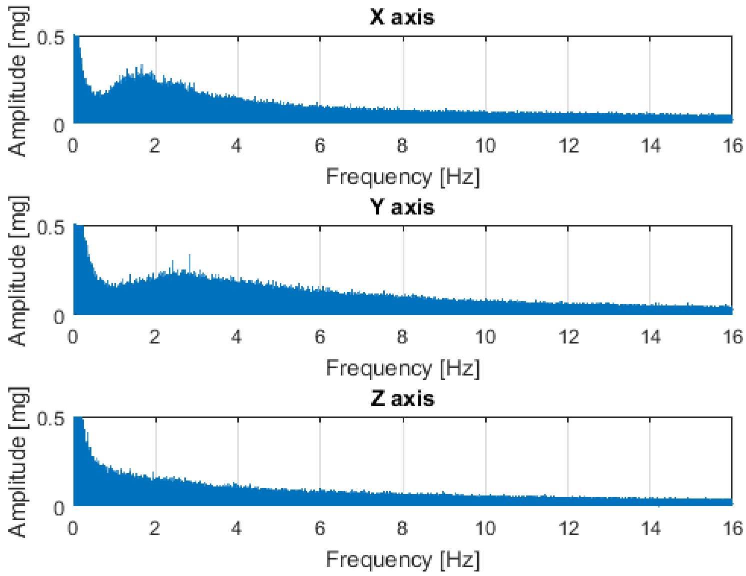

Figure 11: The results of the frequency analysis of the first data set. The frequency spectrum of each axis consists of significant components at frequencies close to zero-frequency and minor components at non-zero frequencies.

closing time was $19.75\,\mathrm{ms}$. The time spent of writing a single data line was from $56\,\mu\mathrm{s}$ to $162.36\,\mathrm{ms}$ and the average was $997.76\,\mathrm{ms}$.

This subsection concentrates in the results of estrus detection. As discussed in section 3.1.3 six cows were used in data recording process. However, two out of six data recordings failed. Thus, that data had to be discarded. The estruses were confirmed by Heatime estrus detection system.

In this study, a total of six cows were used for long period data recording.

4.1.3 Behaviour monitoring

4.2 Algorithm Evaluation

In previous subsection, we discussed of the results in general. The results of the first recorded data set were used for improving the sensor device application software as discussed in section 3.1.2. Conversely, the results of the second data set were used for developing and testing of the algorithms discussed in section 3.3. An additional third set of data was further testing and tuning of the developed algorithms. In this section, we discuss of the results of the estrus detection algorithms. First we discuss of the results of estrus detection in general level. Additionally, we consider the affect of different parameters in estrus detection. Finally, we survey through each algorithm in separate subsections and discuss in detail of their pros and cons.

In order to validate the results of the algorithms, it is mandatory to define some measures of quality. Considering estrus detection, it is rather relevant to define if the cow is in estrus than how much it is in estrus. Therefore, simple categorization of the results such as

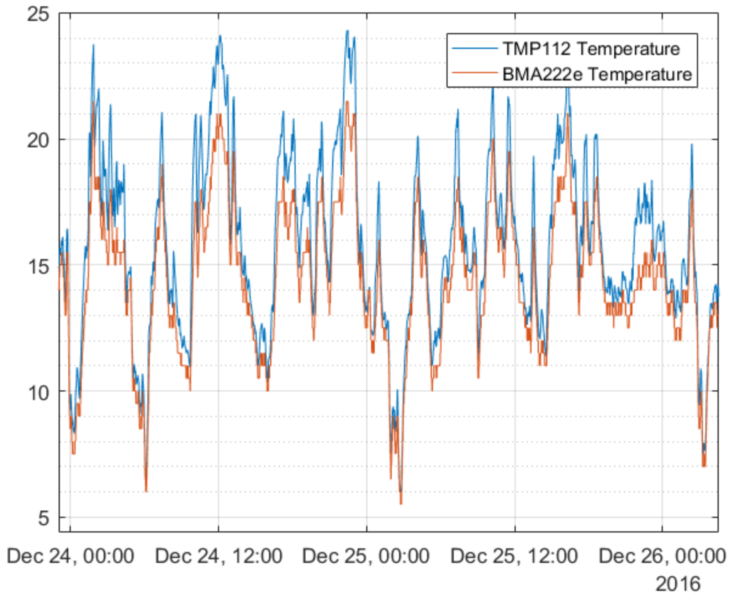

Figure 12: The temperature sensor readings do not correlate with any other results. More likely, temperature correlates with the air temperature inside cowshed.

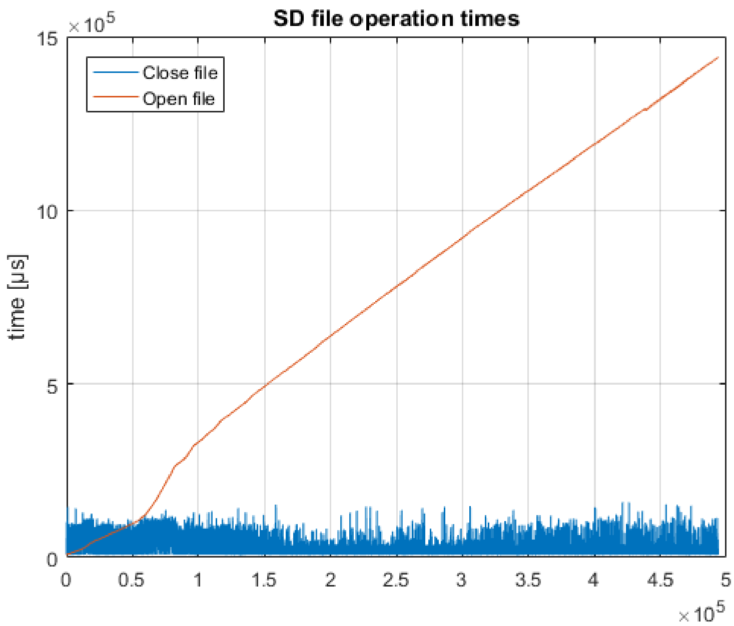

Figure 13:

positive and negative is a fair staring point. Additionally, the positive and negative results shall be divided into true and false results. In conclusion, we have four different measures for the results and they are defined as follows:

- *True positive* means a condition when the algorithm detects an ongoing estrus and

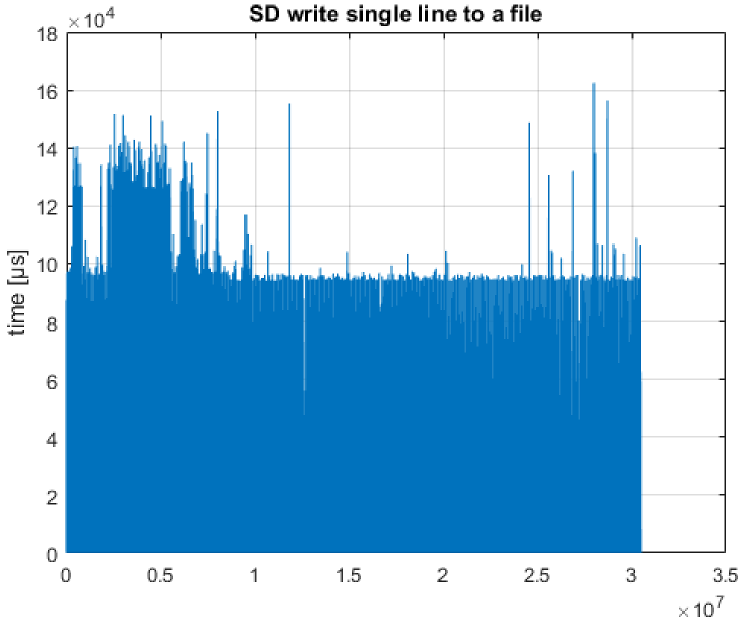

Figure 14:

the cow is in estrus.

- *True Negative* is a condition when the estrus detection algorithm yields negative for estrus and the cow is not in estrus.
- *False positive* states that the estrus detection algorithm yields positive for estrus but the cow is not in estrus.
- *False negative* mean a condition when the estrus detection algorithm yields not in estrus but the cow is in estrus.

In addition to these measures, it is beneficial to be able to depict the differences between positive and negative periods. For example, comparison of the amplitudes. In the following subsections the results are normalized. That is, a median value is subtracted from all the data. Next the data is normalized so that, the extreme value is either ± 1. Therefore, we are able to compare the results within the algorithm as well as between the algorithms.

In this study, the detected estruses are verified with Heatime estrus detection and rumination monitoring system. Thus, each cow wearing the sensor device of this study additionally dons Heatime sensor. The Heatime id considered relatively reliable in estrus detection. Nevertheless, none of the detected estruses were not veterinarian verified. Thus the results of this study are at most as reliable as the results of the Heatime. Furthermore, all the possible false detection of Heatime that conflicts with our device will remain unsolved in this study.

As discussed earlier, a total of six cows were used for the actual estrus data recording. The first three cows were 319, 659 and 9885 of which only two sets of data, 319 and 9885, were valid to use in the algorithm development. Both of the cows had two or more estruses in the data log of the Heatime as shown in pictures A1 and A2.The next three cows were 767, 787 and 812 of which only the data of 767 and 787 were valid to use in algorithm tuning and evaluation. Also both of these cows had detectable estruses in the log of Heatime. They are shown in pictures **??** and A4. Nevertheless, the data recording period in this study lasted only approximately 30 days per instance. Thus, only one detectable estrus per cow were included in our recording period except the cow 9885 with two periods.

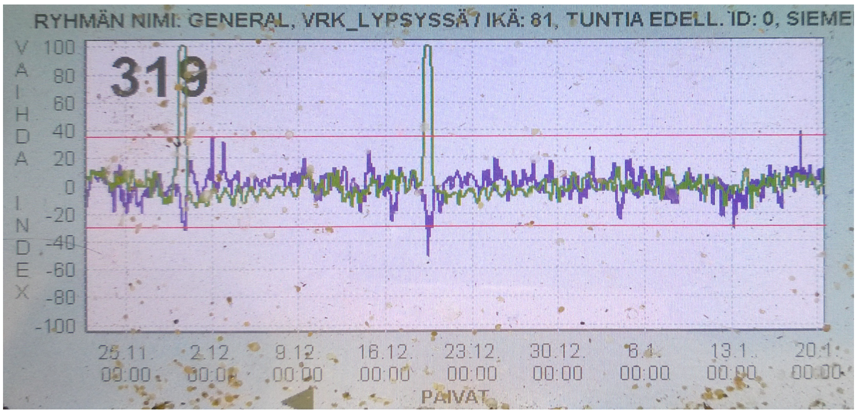

Figure 15: The activity data of Heatime system of the cow 319. The cow has had estruses approximately on October the 30th and December the 20th. The latter of the estruses should be detectable in this study, hence it is within our data recording period.

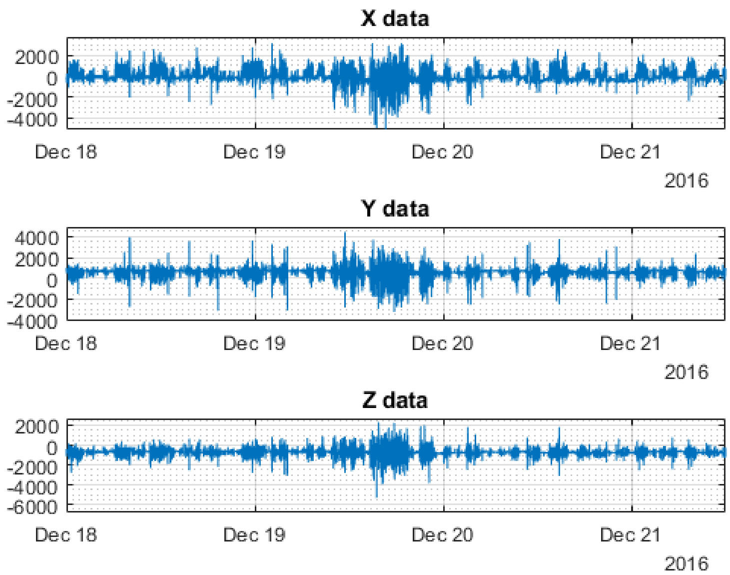

Figure 16: The raw acceleration data of each axis of the cow 319. The time frame of this picture is restricted around the estrus.

The proestrus period corresponding the detected estrus by Heatime is visually detectable in raw accelerometer data. The increased activity level of the cow appears in exceptionally long period of high amplitude activity. Furthermore, the change of activity is detectable in all axis of the accelerometer. Figure 16 represents the raw accelerometer data of the cow 319 around of the proestrus period. Similarly, figures B1 and B2 shows the raw accelerometer data around the both proestrus periods of the cow 9885. In conclusion, these highly active periods are easily detectable for human simply by visually inspecting these plots. Nevertheless, it is necessary to develop computational algorithms that yields either*is in estrus* or *not in estrus*. However, it is the proestrus rather than the actual estrus. Thus, in algorithm level it is more sensible to define whether the cow is in proestrus or is not in proestrus. Consequently, the end of proestrus indicates the beginning of the estrus and this state transition shall be indicated to the end user.

The following subsections discuss of each estrus detection algorithm individually. First we start with the activity measurement algorithm, which is rather straight forwarding and does not require any advantageous computation. Second, we evaluate the variance detection algorithm which is based on variance of periodic short data samples. Lastly, we discuss of the inactivity detection algorithm, which detects rather lack of inactivity than activity directly. Additionally, we discuss of tuning of the algorithm parameters as well as the optimal results based on the recorder data.

The parameters are tuned with iterative methods. That is, testing various values and evaluating the results before applying new values. This iteration is continued until, iteration yields no more benefits.

4.2.1 Activity Monitoring

As stated earlier, data of four different cows were used in the algorithm development and data of two cows had to be discarded since it was not valid. Nevertheless, the data of four cows consists of five proestrus periods, and thus, slightly compensates the smaller amount of usable data.

The first estrus detection algorithm was based on activity measurement. In is very straight forwarding and does not require any complex computation. However, the algorithm is sensitive to offset in the data. Thus, the offset shall be removed with appropriate filter before integrating the values together. Developing of such filter requires knowledge of ...

In despite of the complex algorithm development, the actual filtering requires simple floating point multiplications and summing depending of the order of the filter. Therefore, even with the offset compensating filter this algorithm is rather straight forward and mathematically simple. In contrast to its simplicity this algorithm requires continuous sampling and continuous computing of the samples. First, applying a high-pass filter as an offset compensation. Next, integrating the results. Additionally, there are only few tunable parameters that affect the estrus detection, width of the integration window and threshold. All the other parameters affect only to the computational load and visualization of the results.

According to the duration of proestrus, the optimal integration window is approximately from 9 to 12 hours. Narrower window might trigger the begin on estrus too early. Additionally, it makes the algorithm more sensitive to detect to false positives because the amplitude difference between proestrus and other periods decrease. Therefore, a scared cow behaving restlessly might yield false positive.

Table 1: Increased activity periods detected with Heatime cow monitoring system. Increased activity is considered as proestrus. These detections are used as a reference in this study.

Cow ID	**Occurrence of increased activity**	**Other notifications**
319	30-11-2016	High activity period considered as pro-estrus. However, this period is outside of our sampling period.
319	19-12-2016	Second period of high activity considered as pro-estrus. This activity period is within our data recording period.
9885	24-11-2016	First high activity period of this cow and it is considered as a pro-estrus. However, the occurrence is outside of our data recording period.
9885	22-12-2016	The second activity period of this cow is considered as proestrus. However, this activity period is delayed possibly because of stress cause by the new sensor. Nevertheless, the proestrus is within our data recording period.
9885	04-01-2017	Third high activity period of the cow 9885. In contrast to the previous occurrence, this is slightly early. However, also this occurrence is within our data recording period.
767	21-01-2017	The first high activity period of this cow. Unfortunately, the occurrence is outside of our data recording period.
767	13-02-2017	The second activity period of this cow is within of our data recording period.
787	08-01-2017	The first activity period of this cow. However, the amplitude of the activity is not very high. Therefore, it is considered as proestrus with restrictions. Additionally, the occurrence is outside of our data recording period.
787	19-01-2017	The second increased activity period of the cow. This time the activity is high enough and the event should be considered as proestrus. However, also this occurrence is outside of our data recording period.
787	10-02-2017	The third occurrence of increased activity of this cow. The occurrence is within our data recording period. Thus it can be used as a reference in algorithm development.

The results of the activity measurement algorithm are shown in figures C1, C2, C3 and C4. In these results, three different integration windows are included, 6 hour, 9 hours and 12 hours. In general, the performance of this algorithm is stable. No false positives of false negatives occurred. The difference between proestrus and other periods are obvious. However, the amplitude outside of proestrus is occasionally significant. In addition to the detected estruses, the affect of the width of the integration window is illustrated in picture C5. It is seen that wider window delays the triggering of the end of the proestrus. Conversely, wider integration window enhances the amplitude of the proestrus phase.

Quite straight-forward algorithm without any advantageous data processing. That is, the algorithm integrated the activity in certain periods and represented the activity of the period in one number. Required continuous sampling. Nevertheless, it detected all the estruses easily and no false positives were found.

4.2.2 Variance Detection

Previously we discussed of activity measurement which is the most simple estrus detection algorithm of this study. It was based on continuous filtering and data sampling. Computationally, it was relatively simple but it loads the processor continuously. In contrast, the second estrus detection algorithm, variance detection, is based on samples with regular intervals instead of continuous data streams. Nevertheless, it is computationally slightly more complex than the former algorithm. However, it is still considered as suitable for micro-controller platforms. In addition to higher computational complexity of the algorithm, it comes with two additional tunable parameters, the size of the sample and sampling interval. In the optimized solution of this study, the sample size is 32 data frames and sampling interval is $12\,\mathrm{min}$. According to the configurations of the accelerometer, the 32 data frames responses to approximately 2 second period of data.

Figures D1, D2, D3 and D4 represents the results with optimized parameters. The parameters for all algorithms are common. As seen, the algorithms occationally detects the proestrus. However, the results includes numerous false positives and even one false negative with the cow 9885. The reliability, seems to depend on the cow and individually tuned parameters. Nevertheless, no appropriate parameters were found for all f the cows during this study.

The data of the cow 9885 were analyzed with various parameters because the algorithm yielded one true positive and one false negative. Thus it was interested to tune the parameters and see if the algorithm could yield two true positives.

Varying the sample size affect the results:

4.2.3 Inactivity Detection

Similarly to the variance detection algorithm, the inactivity detection algorithm has to more tunable parameters than the activity monitoring algorithm. Nevertheless, the parameters differ from the previous algorithms. That is, the tunable parameters are the motion threshold and the delay the motion shall not exceed the threshold. Otherwise, the algorithms follows similar integration time as the activity monitoring algorithm. Additionally, testing showed that this algorithm is not very sensitive to the parameters. Consequently, proestruses were detected as true positives with any feasible parameters. Furthermore, the algorithm did not yield any false positives or false negatives during this study.

The first set of data provided relatively wide frequency spectrum of data.

The sampling period provided a total of three probable estruses. One of the sensors failed during the sampling period, and therefore, no estrus could be estimated. However, one of the cows had two estruses during the sampling period and thus, compensated the failed sensor.

Based on pure visual analysis of the acceleration data, the red cow has two highly active proestrus periods. First begins December the 22nd at 9 pm end ends December the 23rd at 9 am. The second period begins January the 4th at 2 am and ends 6 pm. Proper insemination is from 6 to 24 hours after proestrus period.

The estrus was detected using Heatime estrus detection system. The detected estruses are represented in figures...

The activity could be measured in several ways. Pedometers measures the number of steps. However, sensor worn in neck could not probably be used for step counting and therefore, no steps were estimated. Human activity bracelets and other devices could measure steps but if total energy consumption is being estimated, the values of acceleration should be taken into account. A research have shown that the sum of absolute value of the axis, instead of a absolute value is the best simple estimate for energy consumption and therefore, a good activity measure.

The cow activity was measured as a sum of absolute value acceleration vector. The activity was summed over a varying time window from 6 hours to 36 six hours. Since the resulting activity is always a sum of past activity, this method causes delay and widening of the activity peak. However, the size of the window does not affect the steepness of the rising or falling edges. Nevertheless, the size of the window widens the peaks if the window size is too high. In result, increasing the window might delay the estrus detection, if the detection has dependencies in the falling edge. Based on the three found heat activity peaks of estrus, an optimal window size is somewhere between 9 end 12 hours.

4.3 Conclusions

Previously in this section we discussed of the results of the research of this study. First, we discussed the results in general level. Next we focused on presenting the results of the estrus detection algorithms. Consequently, in this subsection we will draw conclusions based on the presented results. First, we will evaluate sensor based estrus detection as a concept. The evaluation will take into account former studies as well as the results gained in this study. Second, We will focus on failures of this study and propose actions to correct the failures and mistakes. Third, the scope is in suggestions for the future studies as continuation for this study.

4.3.1 Sensor Based Estrus Detection

This study introduced a sensor device for dairy cow estrus detection. The introduction included as well as the hardware configuration as well as the software implementation. Furthermore, we discussed of three different estrus detection algorithms suitable for the device. Practically, the algorithms were developed and tested with Matlab on computer after recording of real cow data. The algorithms were different in their approach to the topic. However, all of them were based on the accelerometer data only. In general, all of the algorithms performed the estrus detection well after tuning of the algorithm specific parameters. Nevertheless, there were differences in the reliability as well as in the clearance of the results. The conclusions of the algorithms are discussed in following.

The activity monitoring algorithm was a straightforward integration of the accelerometer data. The only tunable parameter was the size of the integration time. The most suitable integration time corresponds approximately the duration of the proestrus phase of the estrus cycle, that is roughly from 9 to $12\,\mathrm{h}$. Extending of the integration time only delayed the triggering of the begin of estrus. Thus, it was not beneficial. Conversely, integration time less than $6\,\mathrm{h}$ yielded a risk of triggering estrus too early in the middle of the proestrus phase. These results were so obvious, hence, it was decided to use $9\,\mathrm{h}$ window with the rest of the algorithms without further tuning of it. Furthermore, the algorithm was quite robust. That is, the results were clear and there were no high risk of yielding false results within the suggested integration time. However, the algorithm is sensitive to data offset and it shall be removed before any other computation. Nevertheless, offset did not prevent from detecting the estrus in this study. Actually, it only reduced the amplitude difference between the proestrus and other periods.

In contrast to the activity monitoring algorithm, the variance detection algorithm was sensitive to the parameters. That is, too short sample or too long sampling interval effected to the results significantly. However, ...

Similarly to the variance detection, the inactivity detection algorithm had two additional tunable parameters. However, inactivity detection did not seem to be sensitive to the tuning of the parameters. In this study, any reasonable parameter values yielded true positive and true negative results. Furthermore, no false results occurred. Nevertheless, the most radical parameter values begun to affect to the amplitude difference between proestrus and other phases of the estrus cycle. In conclusion, the performance and reliability of this algorithm seemed most promising in this study. Furthermore, this algorithm is the most suitable for the micro-controller based solutions. That is, most of the computations are performed in the accelerometer and the micro-controller only summarizes seldom interrupt the accelerometer generates. Additionally, it enables efficient use of power saving modes of the micro-controller which is critical with battery powered solutions.

The dairy cow estrus is detectable with triaxial accelerometer only and no other sensors are required. The results were verified with one single commercial product. Thus, the results may not be completely reliable and valid. Furthermore, none of the detected estruses were verified by a vet.

The estrus detection algorithms developed in this study are suitable for micro-controller environments. Therefore, algorithm computation with computers is not required. Consequently, there is no need to large data storage or transmitting raw sensor data wirelessly.

4.3.2 Failures and considerations

Previously, we discussed of sensor based dairy cow estrus detection based on the results of this study. The conclusion was that the estrus of a dairy cow is detectable with an accelerometer base wearable sensor. Furthermore, it seemed the estrus is rather detectable with various algorithms. However, there were differences in the performance and reliability of different algorithms. Whereas the previous subsection focused on the successful part of this study, in this subsection the scope is in failures and suggestions for improvements.

Firstly, the body temperature measurement was a total failure. That is, the heat conducting system did not conduct the heat enough. Conversely, the results of the temperature measurements seemed to follow the temperature of the surrounding air. It was also seen from the temperature curve of the accelerometer 12, hence, both of the temperatures seemed synchronous. Additionally, it was tested to filter the results and calculate the differential of the two temperatures sensors. Nevertheless, no correlation with a ongoing estrus or proestrus were found in this study. In contrast to the sensor setup of this study, the temperature sensor should be placed into more direct skin contact. Thereby, the sensor would more likely measure the body temperature rather than the surrounding air. Nevertheless, this discussion is only hypothetical. Furthermore, the body temperature was assumed to rise only up to two degrees during the estrus. Therefore, detecting of such minor alterations with micro-controller suitable algorithms may not be reliable at all.

Secondly, total of three attempts to record cow data were ruined because of an error state of the sensor device. Unfortunately, there was no method to define the cause of the error directly. However, it seemed that the device continued working even in the error state but the data was invalid. However, further analysis of the error data provided an assumption that the accelerometer had reset meanwhile the micro-controller had not. Consequently, the micro-controller stopped receiving any interrupts from the accelerometer. Furthermore, the range of the accelerometer was set to default of 2 g. Meanwhile, the micro-controller applications continued of executing of its software loop. Nevertheless, without fetching data normally. Therefore in both software implementations, the watchdog timer triggered an event for reading single data frame from the FIFO memory of the accelerometer. Considering the number of the invalid samples and the period of watchdog timer matched with the time cow was wearing the sensor. In order to avoid such failures in the future, the software should include a method for detecting erroneous state of the accelerometer. Additionally, the hardware might be causing blackouts to the accelerometer. This possibility should be inspected and the hardware design improved according the discoveries.

Lastly, the hardware design did not provide optimal performance for high sampling rate as seen in section 4.1. Therefore, the first recorded set of data was partially corrupted because of frequent buffer overflows. Furthermore, high sampling rate fulfilled the SD memory card relatively fast. Thus high sampling rate together with SD memory card would not be preferred solution in long term data recording.

4.3.3 Suggestions for Future Studies

This study has introduced three different sensor based algorithms for dairy cow estrus detection. Furthermore, we have evaluated the algorithms and concluded that the estrus id detectable with accelerometer based wearable sensor. Additionally, we have discussed of secondary products and other observations during this study. Lastly in this study, we will discuss of suggestions for potential future studies based on the introduced results and conclusions.

Firstly, the detected estruses in this study were not confirmed at any point. The results were verified only with another sensor device as a reference. Thus, the results of this study are not completely reliable. Therefore, the algorithms developed in this study shall be implemented directly for micro-controller. Furthermore, they shall be tested with dairy cattle in real-time instead of posterior data processing sessions. In order to fully validate the algorithms, the results shall be verified by a veterinary. That is, each cow shall be inseminated once an estrus has been detected. Consequently, the detection is considered as true positive if the insemination is successful and the cow becomes pregnant. Nevertheless, the normal success rate of insemination as well as the estrus cycle shall be taken into account.

Secondly, the amount of data used for algorithm development was quite small. That is, only four months of cow data was valid to use in algorithm development. Luckily, the data consisted total of five estruses. Nevertheless, statistically sampling is minor considering the head count world wide. Furthermore, it is assumed there are significant differences in cow activity levels, hence, it varied already in this study. Additionally, a complementary study should cover various breeds in various environments e.g., tie-stall and pasture. Furthermore, we did no discuss of algorithm tuning thoughtfully. Actually, the methods of this study tuned the algorithms for each cow individually after analyzing all of the data. Thus, the parameters suitable for one cow might not be suitable for another. However, more of recorded data would enable the developing of algorithms with self-adjustable parameters. Consequently, it would improve the real-time performance of the estrus detection algorithms discussed in this study.

Thirdly, behavior monitoring was considered as on optional topic in this study. A successful behavior monitoring would provide prominent information for the farmer about the state of the cattle. Furthermore, exceptions in behavior could indicate concerns in the cows health and treatment could be started before issues escalate. In this study, we recorded in-cowshed video for studying of behavior recognition. However, the the video quality was insufficient and there were no valid time synchronization between the video and motion data. Thus, this option was discarded in early phase of this study. However, based on the brief preliminary study, this option seemed promising. Therefore, study of valid motion and video data with proper time synchronization would provide algorithms for such topics as rumination and lameness detection.

5 Summary

Summary of all the previous. Alternatively “Discussion” or “Conclusions”...

References

[1] June 24, 2013. URL: https://www.proagria.fi/ajankohtaista/lehmien-tuotos-kasvaa-jokaisella-poikimakerralla-804 (visited on 01/19/2017).

[2] United States Department of Agriculture. *Agricultural Statistics 2015*. URL: https://www.nass.usda.gov/Publications/Ag_Statistics/2015/Ag_Stats_2015_complete%20publication.pdf (visited on 01/18/2017).

[3] United States Department of Agriculture. *Livestock and Poultry: World Markets and Trade*. Foreign Agricultural Service. Oct. 2016. URL: http://usda.mannlib.cornell.edu/usda/fas/livestock-poultry-ma//2010s/2016/livestock-poultry-ma-10-12-2016.pdf (visited on 01/18/2017).

[4] L. Mattias Andersson et al. "Wearable wireless estrus detection sensor for cows". In: *Computers and Electronics in Agriculture* 127 (2016), pp. 101–108. ISSN: 0168-1699. DOI: http://dx.doi.org/10.1016/j.compag.2016.06.007. URL: http://www.sciencedirect.com/science/article/pii/S0168169916303581.

[5] L. M. Andersson et al. "Wearable wireless sensor for estrus detection in cows by conductivity and temperature measurements". In: *SENSORS, 2015 IEEE*. Nov. 2015, pp. 1–4. DOI: 10.1109/ICSENS.2015.7370219.

[6] Arduino. URL: https://www.arduino.cc/en/Main/Software (visited on 09/16/2017).

[7] Arduino. URL: https://www.arduino.cc/en/Reference/Wire.

[8] Arduino. URL: https://www.arduino.cc/en/Reference/SPI.

[9] Arduino. URL: https://www.arduino.cc/en/Reference/SD (visited on 09/23/2017).

[10] *ATmega16U4/ATmega32U4 8-bit Microcontroller with 16/32K bytes of ISP Flash and USB Controller*. Rev. 7766J – 04/2016. Atmel Corporation. Aug. 2007.

[11] *BMA222E Digital, triaxial acceleration sensor*. BST-BMA222E-DS004-6. Rev. 1.3. Bosch. Apr. 2005.

[12] Leandro dos Anjos Brunassi et al. "Improving detection of dairy cow estrus using fuzzy logic". en. In: *Scientia Agricola* 67 (Oct. 2010), pp. 503–509. ISSN: 0103-9016. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-90162010000500002&nrm=iso.

[13] Heli Castrén. *Kotieläinten käyttäytyminen ja hyvinvointi*. Mikkeli: Helsingin yliopiston Maaseudun tutkimus- ja koulutuskeskus, 1997.

[14] *Digital Filters*. 2nd ed. Springer-Verlag Berlin Heidelberg, 2011. ISBN: 978-3-642-14325-0. DOI: 10.1007/978-3-642-14325-0.

[15] Eric Dufresne. Mar. 20, 2005. URL: https://upload.wikimedia.org/wikipedia/commons/d/db/Cows_on_a_farm_-_by_Eric_Dufresne.jpg (visited on 08/26/2017).

[16] GEA Farm Technologies GmbH. URL: http://www.gea.com/en/binaries/DairyFarming_CowScout_Brochure_EN_0315_tcm11-21065.pdf (visited on 10/21/2017).

[17] *Heatime HR User Guide*. Rev. 2.13.X.X. SCR Engineers LTD. July 2014. URL: http://www.heatime.dk/wordpress/wp-content/uploads/2012/07/HeatimeHR-User-Manual-2.13.X.X.pdf.

[18] Jan Hulsen. Ed. by Juho Kyntäjä. 1079. Vantaa: ProAgria keskusten liitto, 2014. ISBN: 978-951-808-192-3.

[19] R. Jónsson et al. "Oestrus detection in dairy cows from activity and lying data using on-line individual models". In: *Computers and Electronics in Agriculture* 76.1 (2011), pp. 6–15. ISSN: 0168-1699. DOI: http://dx.doi.org/10.1016/j.compag.2010.12.014. URL: http://www.sciencedirect.com/science/article/pii/S0168169910002656.

[20] Dong-Won Kang et al. "Prediction of energy consumption according to physical activity intensity in daily life using accelerometer". In: *International Journal of Precision Engineering and Manufacturing* 13.4 (2012), pp. 617–621. ISSN: 2005-4602. DOI: 10.1007/s12541-012-0079-2. URL: http://dx.doi.org/10.1007/s12541-012-0079-2.

[21] A.A. Khan. *Digital Signal Processing Fundamentals*. Charles River Media Computer Engineering. Da Vinci Engineering Press, 2005. ISBN: 9781584502814. URL: https://books.google.fi/books?id=%5C_xeQNRlxzG4C.

[22] Tarja Kortesmaa and Aimo Jokela, eds. *Ruoka- ja luonnonvaratilastojen e-vuosikirja 2016*. Luonnonvara, ja biotalouden tutkimus 64/2016. Luonnonvarakeskus, 2016. ISBN: 978-952-326-322-2.

[23] S. López et al. "On the analysis of Canadian Holstein dairy cow lactation curves using standard growth functions". In: *Journal of Dairy Science* 98.4 (2015), pp. 2701–2712. ISSN: 0022-0302. DOI: http://dx.doi.org/10.3168/jds.2014-8132. URL: http://www.sciencedirect.com/science/article/pii/S0022030215000466.

[24] IceRobotics Ltd. URL: http://www.icerobotics.com/wp-content/uploads/2017/05/IceRobotics_Product_Brochure_2017.pdf (visited on 10/21/2017).

[25] Moocall Ltd. URL: https://moocall.com/products/moocall-sensors (visited on 10/21/2017).

[26] Well Cow Ltd. URL: http://wellcow.co.uk/bolus/ (visited on 10/21/2017).

[27] Paula Martiskainen et al. "Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines". In: *Applied Animal Behaviour Science* 119.1–2 (2009), pp. 32–38. ISSN: 0168-1591. DOI: http://dx.doi.org/10.1016/j.applanim.2009.03.005. URL: http://www.sciencedirect.com/science/article/pii/S0168159109000951.

[28] MathWorks. URL: https://se.mathworks.com/products/matlab.html#overview (visited on 09/17/2017).

[29] G.J. Miao. *Signal Processing in Digital Communications*. Norwood, 2007.

[30] M. Pastell et al. "A wireless accelerometer system with wavelet analysis for assessing lameness in cattle". In: *Biosystems Engineering* 104.4 (2009), pp. 545–551. ISSN: 1537-5110. DOI: http://dx.doi.org/10.1016/j.biosystemseng.2009.09.007. URL: http://www.sciencedirect.com/science/article/pii/S1537511009002748.

[31] Anne Penninkangas. "Nautojen hoitoympäristö". In: *Kotieläintilan luomuopas*. Tieto tuottamaan 78. Maaseutukeskusten Liitto, 1998. ISBN: 951-808-064-X.

[32] Pixabay. Mar. 31, 2016. URL: https://pixabay.com/fi/lehm%C3%A4-farm-pilttuu-el%C3%A4imet-1318283/ (visited on 08/26/2017).

[33] K.R. Rao, D.N. Kim, and J.J. Hwang. *Fast Fourier Transform - Algorithms and Applications*. Signals and Communication Technology. Springer Netherlands, 2012. ISBN: 9789048118892. URL: https://books.google.fi/books?id=i9P3sgEACAAJ.

[34] Mike Schwager. URL: https://github.com/GreyGnome/EnableInterrupt.

[35] *SD Specifications Part 1 Physical Layer Simplified Specification*. Version 6.00. SD Association. Apr. 2017.

[36] Raimo Seppänen and Seppo Tiihonen. "Matematiikka". In: *MAOL-taulukot*. 9th ed. Helsinki: Kustannusosakeyhtiö Otava, 1991.

[37] *SPI Block Guide*. Rev. 03.06. Motorola, Inc. Feb. 2003. URL: https://web.archive.org/web/20150413003534/http://www.ee.nmt.edu/~teare/ee308l/datasheets/S12SPIV3.pdf.

[38] Li Tan. *Digital Signal Processing: Fundamentals and Application*. Burlington: Elsevier Science, 2007. ISBN: 9.

[39] ThoughtCo. *Dairy Farming - The Ancient History of Producing Milk*. Feb. 9, 2017. URL: https://www.thoughtco.com/dairy-farming-ancient-history-171199 (visited on 08/19/2017).

[40] *TMP112 High-Accuracy, Low-Power, Digital Temperature Sensor With SMBus and Two-Wire Serial Interface in SOT563*. SBOS473E. Rev. E. Texas Instruments. Mar. 2009.

[41] *UM10204 I^2C-bus specification and user manual*. Rev. 6. NXP Semiconductors. Apr. 2014. URL: https://www.nxp.com/docs/en/user-guide/UM10204.pdf.

[42] Jorge A. Vázquez Diosdado et al. “Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system”. In: *Animal Biotelemetry* 3.1 (2015), p. 15.

[43] Eeva Vornanen. “Naudan hoitoympäristö”. In: *Luomunaudan ja -sian ruokinta ja hoito*. Tieto tuottamaan 94. Maaseutukeskusten Liitto, 2001. ISBN: 951-808-097-6.

[44] Aaron Wisner. URL: https://github.com/wizard97/ArduinoRingBuffer (visited on 09/23/2017).

A Heatime Data Plots

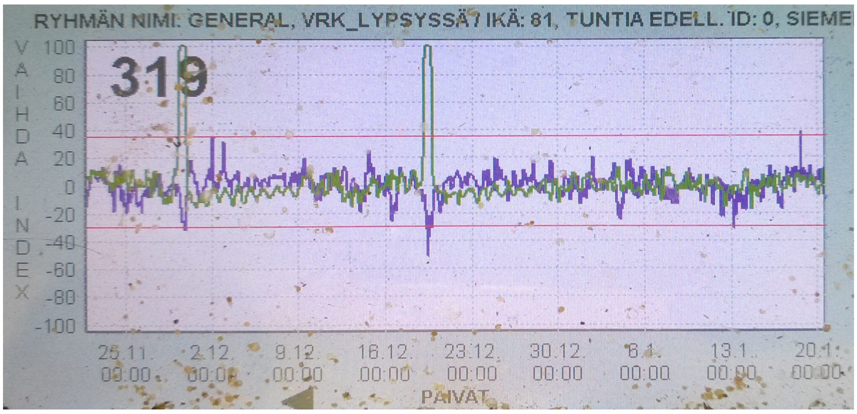

Figure A1: The activity data of Heatime system of the cow 319. The cow has had estruses approximately on October the 30th and December the 20th. The latter of the estruses should be detectable in this study, hence it is within our data recording period.

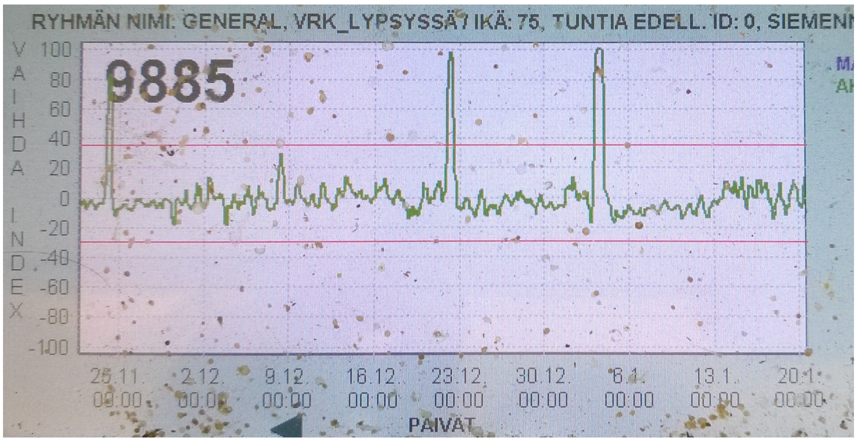

Figure A2: The activity data of Heatime system of the cow 9885. The cow has had estruses approximately on October the 24th, December the 22nd and January the 4th. Two latter estruses should be detectable in this study, hence they both are within our dat recording period.

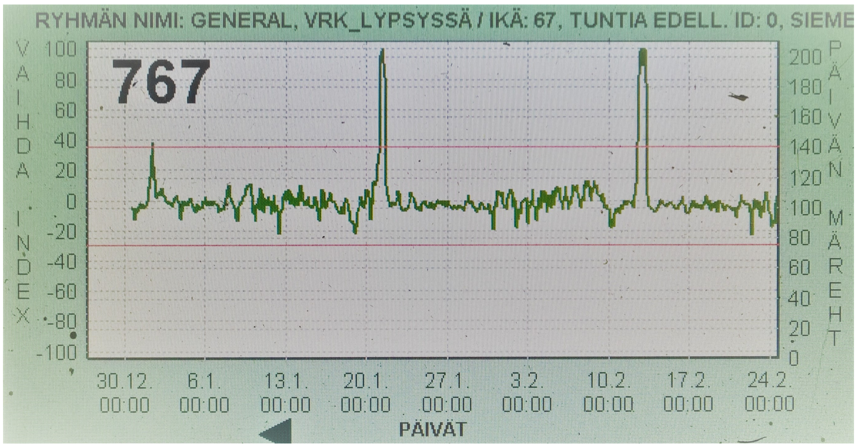

Figure A3: The activity data of Heatime system of the cow 767. The cow has had estruses approximately on January the 21st and February the 13th. The latter of the estruses should be detectable in this study, hence it is within our data recording period.

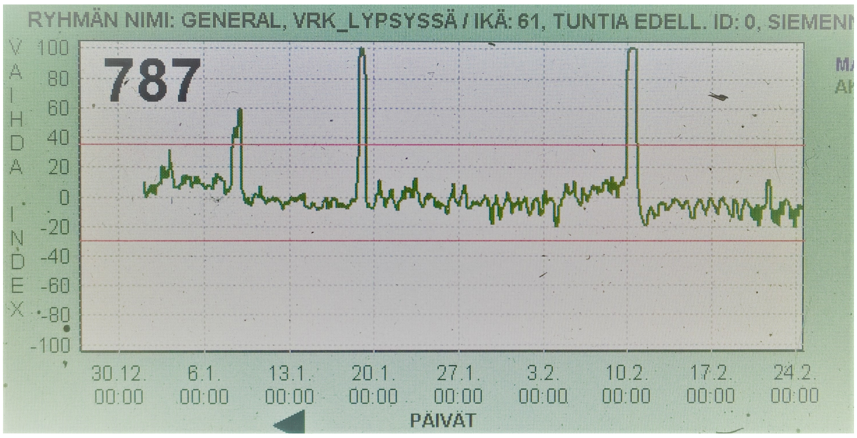

Figure A4: The activity data of Heatime system of the cow 787. The cow has had estruses approximately on January the 19th and February the 10th. The latter of the estruses should detectable in this study, hence it is within our data recording period.

B Raw Accelerometric Data

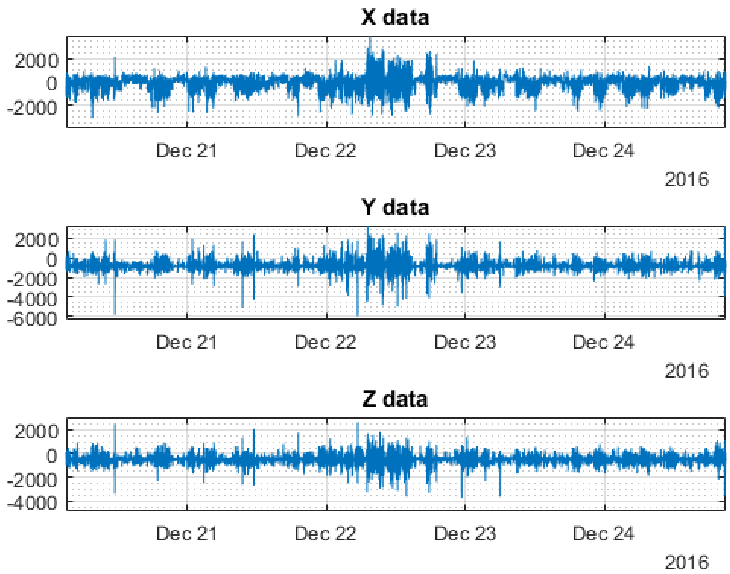

Figure B1: The raw acceleration data of each axis of the cow 9885. The time frame is scaled around the first detectable estrus period.

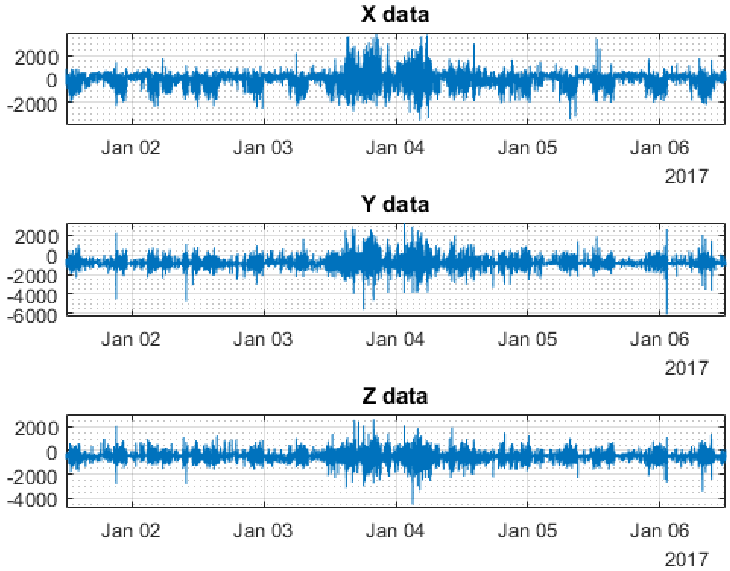

Figure B2: The raw acceleration data of each axis of the cow 9885. The time frame is scaled around the second detectable estrus period.

C Activity Monitoring Full Results

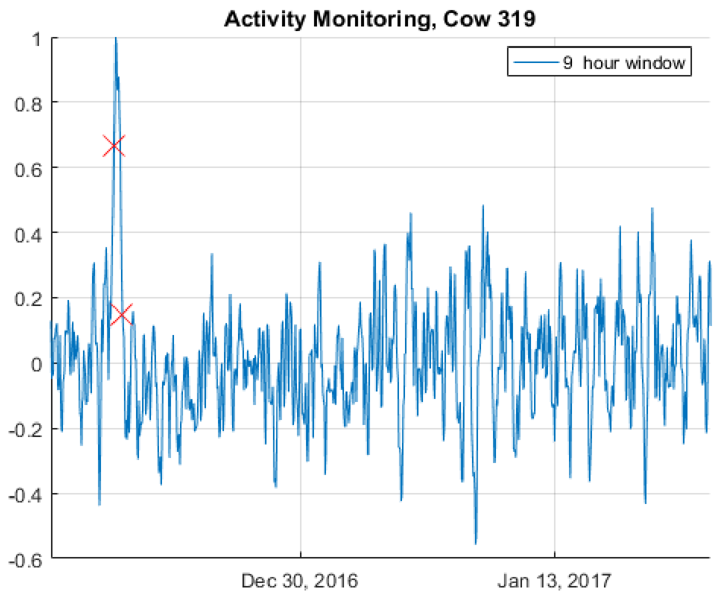

Figure C1: The plot of the results of the activity measurement algorithm of cow 319. A true positive estrus is detected and no false positive or false negative detection occurred. However, there is a lot of variation when not in estrus. Thus, the risk of false positive and false negative results is plausible.

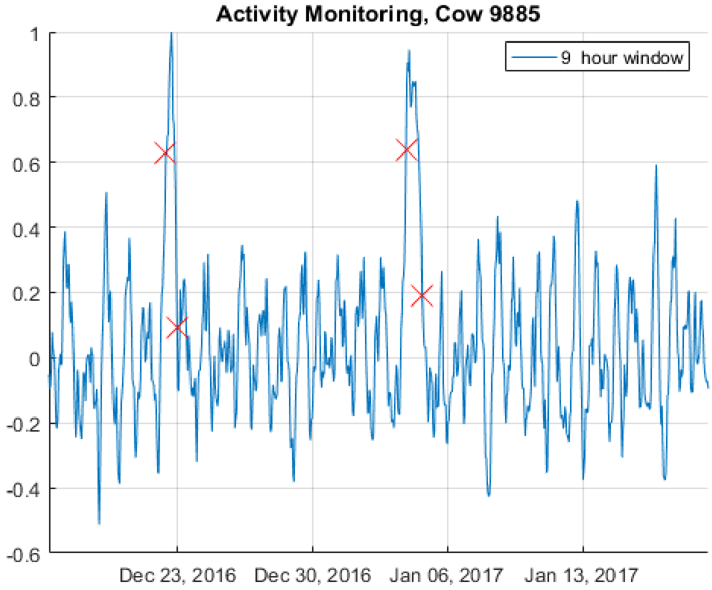

Figure C2: The results of activity measurement of the cow 9885. Both of the estruses are detected. However, the difference between the estruses and the rest of the period is insignificant as it was with cow 319. Thus, the possibility of false positive and false negative results exists.

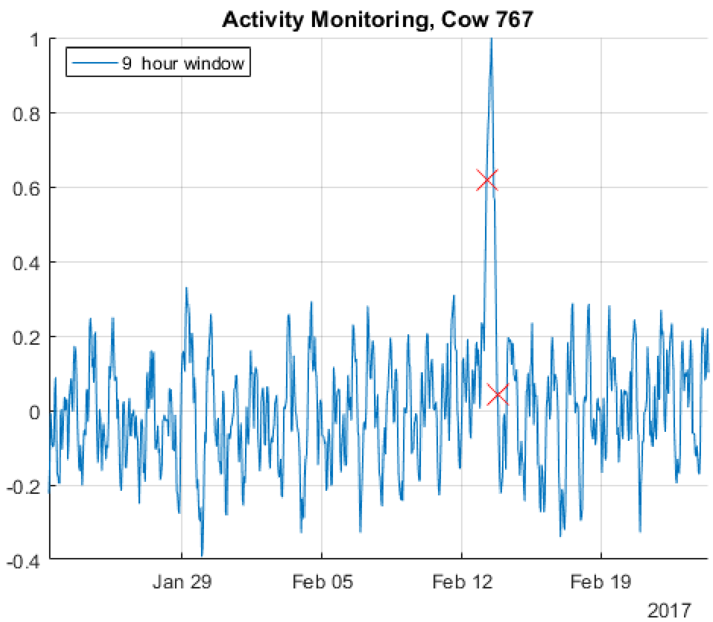

Figure C3: The plot of the activity measurement results of the cow 767. A true positive estrus is detected and no occurrence of false positives of false negatives. Additionally, the difference between proestrus and rest of the period is obvious. The risk of false positive or false negative results is minor.

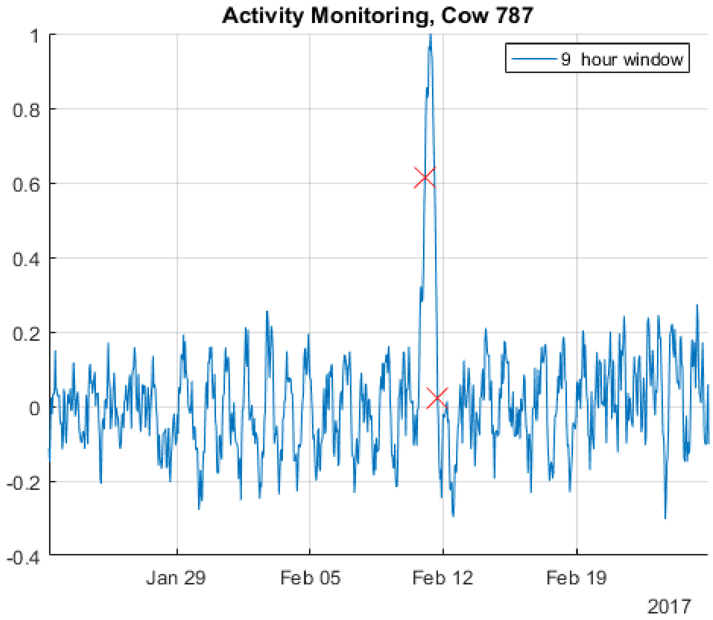

Figure C4: The results of activity measurement of the cow 787. The difference between the estrus and rest of the period is most distinct within this algorithm. Thus, the risk of false positive and false negative results is least significant.

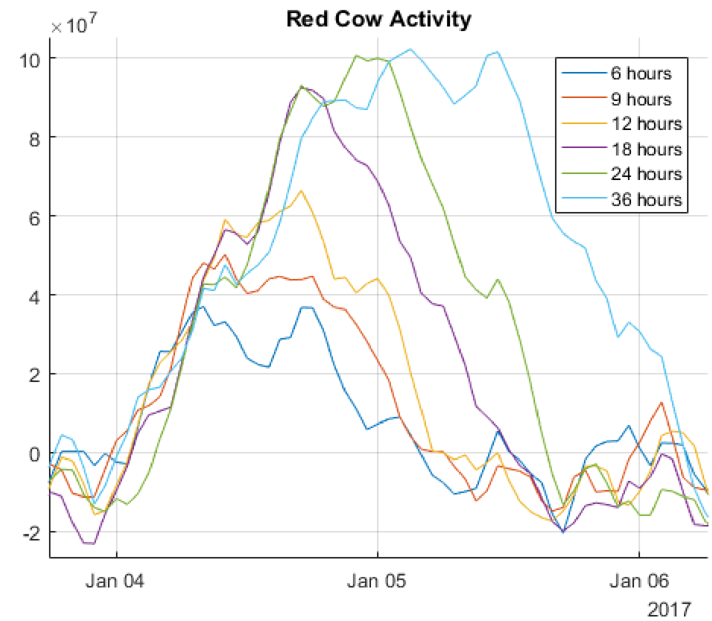

Figure C5: The activity plots of the second proestrus of the cow 9885. The figure illustrates the affect of varying the size of the integration window. Wider window increases the amplitude of the estrus and eases the detectability. However, simultaneously it delays the moment of detection.

D Variance Detection Full Results

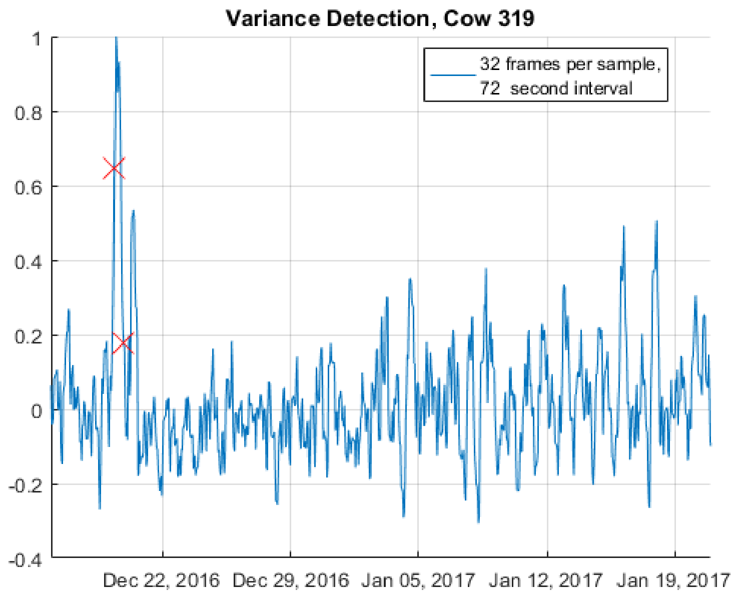

Figure D1: The results of the variance detection algorithm of the cow 319. The estrus on December the 20th is barely detectable. Additionally, algorithm yielded several false positive estruses. Furthermore, the amplitude of the false positives exceeds the true positive.

Varying the sampling interval affect the results:

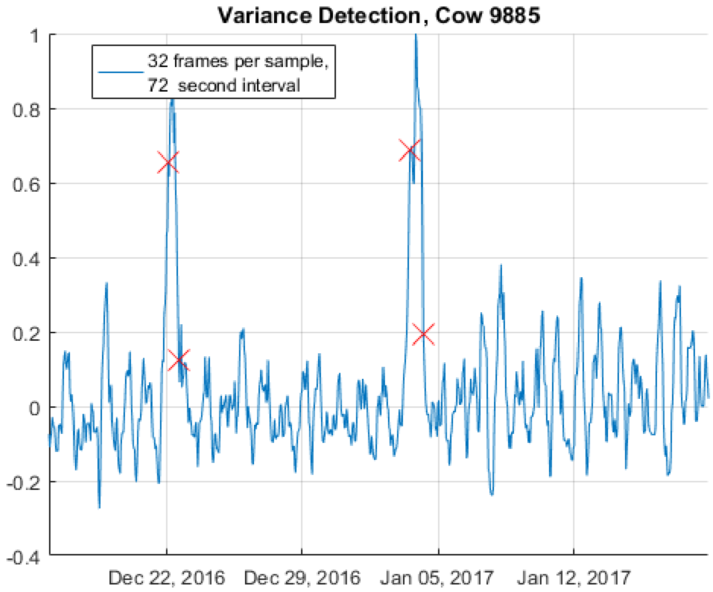

Figure D2: The results of variance detection algorithm of the cow 9885. The true positive estrus is detected on December the 22st. Furthermore, the amplitude difference between the detected estrus and other period is significant. However, the algorithm yield a false negative on January the 4th.

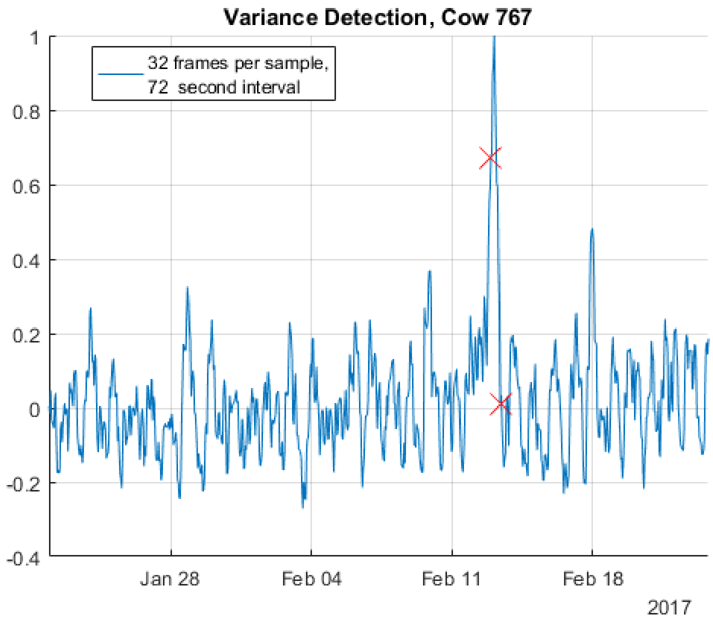

Figure D3: The variance detection results of the cow 767. There are multiple false positive detections in the data period. In general, there is no obvious difference between the estrus and non-estrus periods. Nevertheless, a true positive estrus is detected on February the 13th.

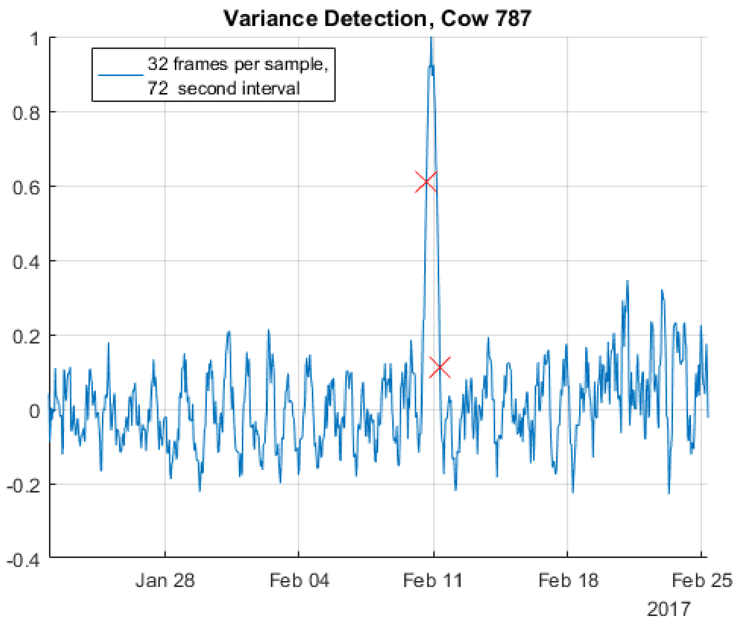

Figure D4: The results of variance detection algorithm of the cow 787. The true positive estrus is detected on February the 10th. However, a false positive detection occurred on February the 21st. Otherwise, the amplitude difference between estrus and non-estrus periods is obvious.

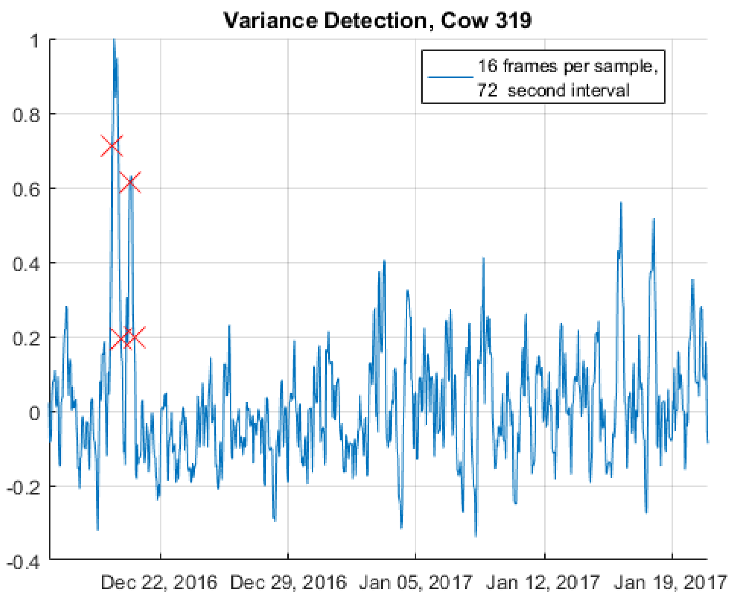

Figure D5: Reducing the sample size to 16 frames per sample and keeping the sampling interval in 72 seconds yields false positive results with the cow 319

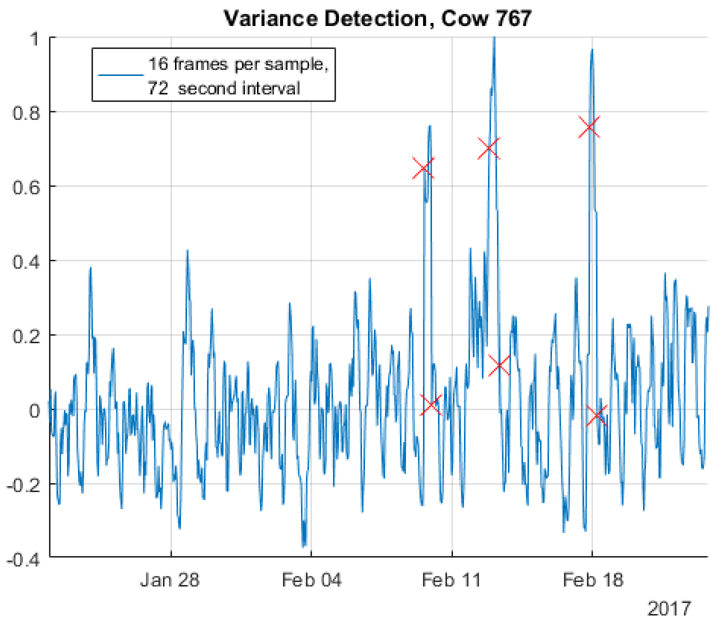

Figure D6: Reducing the sample size to 16 frames per sample and keeping the sampling interval in 72 seconds yields false positive results with the cow 767

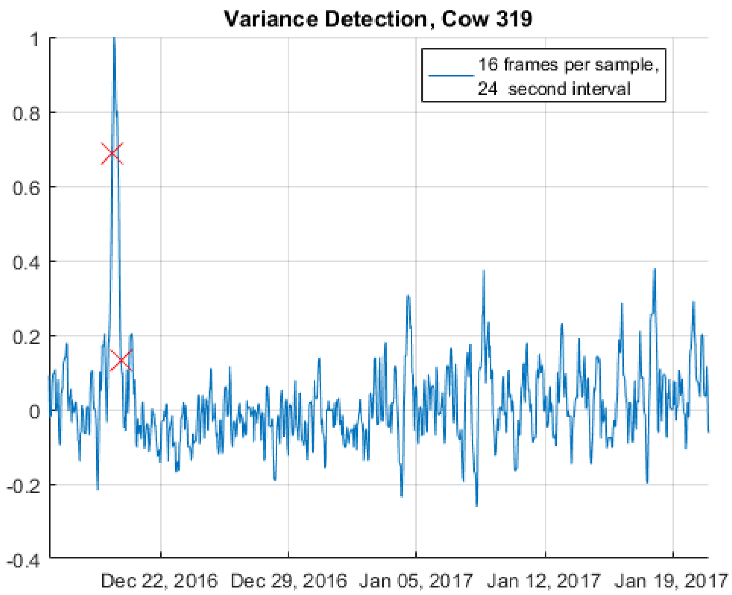

Figure D7: Reducing the sample size to 16 frames and increasing the sampling interval to 24 seconds improves the performance of the variance detection algorithm. Consequently, no false positive results occurs.

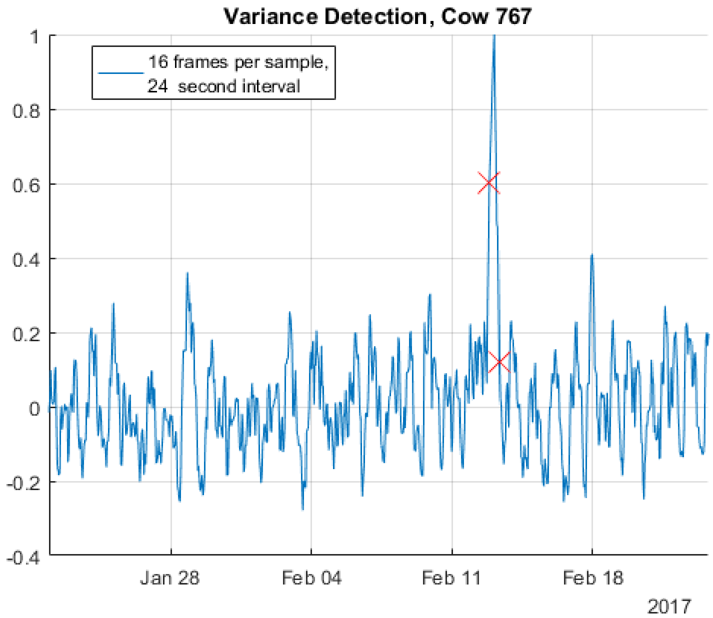

Figure D8: Reducing the sample size to 16 frames and increasing the sampling interval to 24 seconds improves the performance of the variance detection algorithm. Consequently, no false positive results occurs.

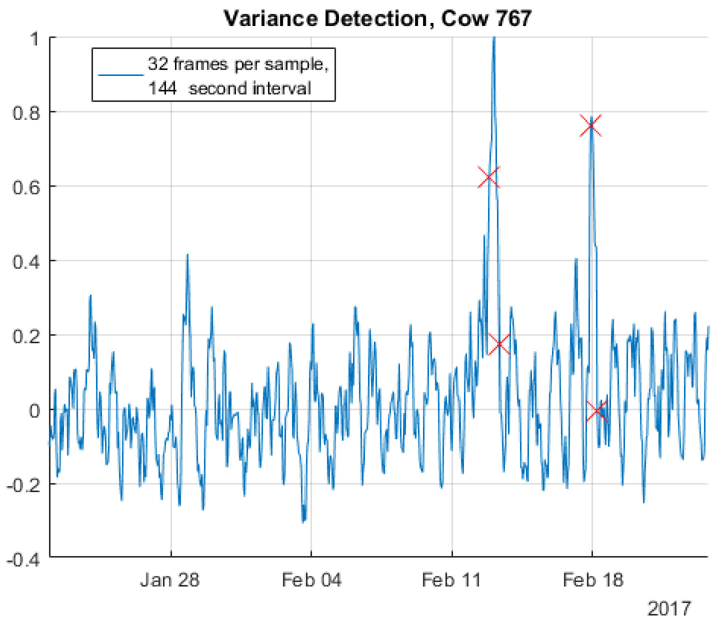

Figure D9: More seldom sampling yield false positive results as seen in this picture.

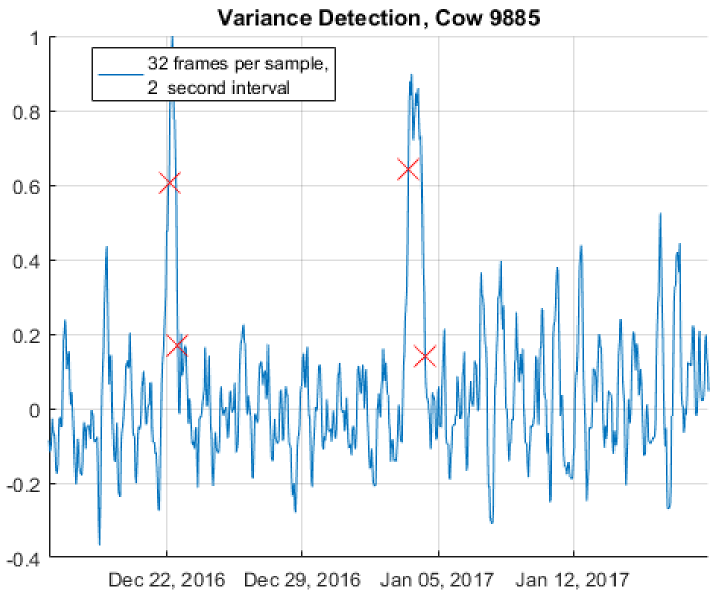

Figure D10: Decreasing the sampling interval does not directly improve the results as it is with the cow 9885. This sampling frequency corresponds approximately continuous sampling. Nevertheless, the results are worse than formerly.

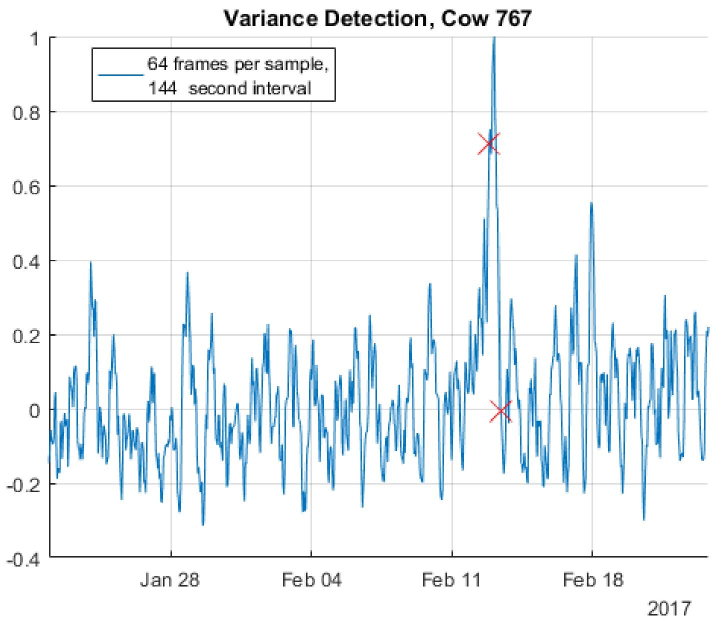

Figure D11:

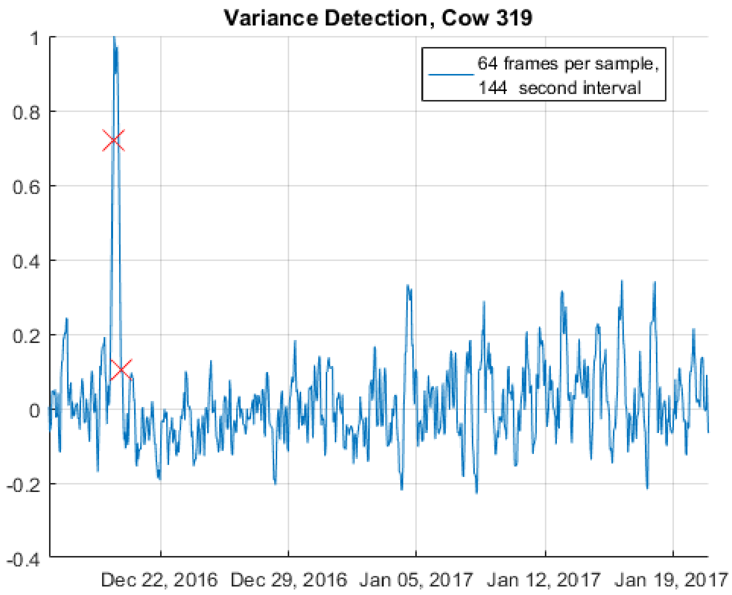

Figure D12:

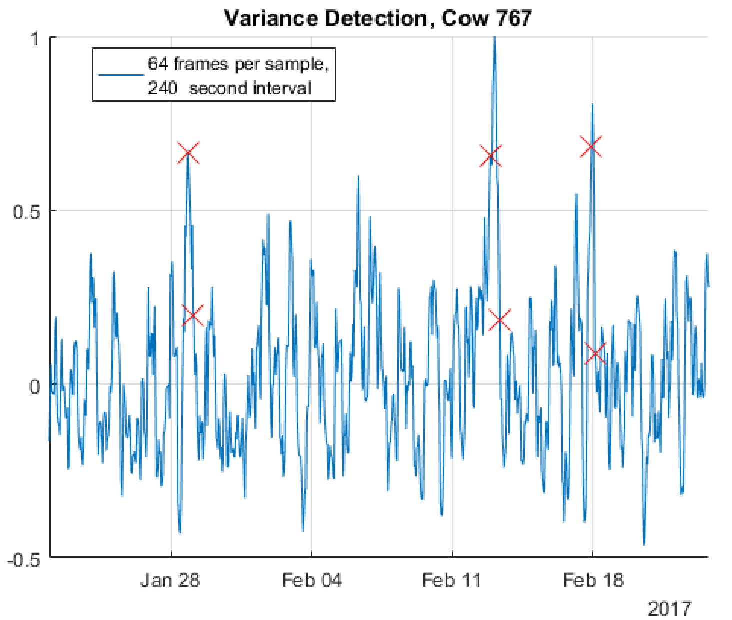

Figure D13:

E Inactivity Detection Full Results

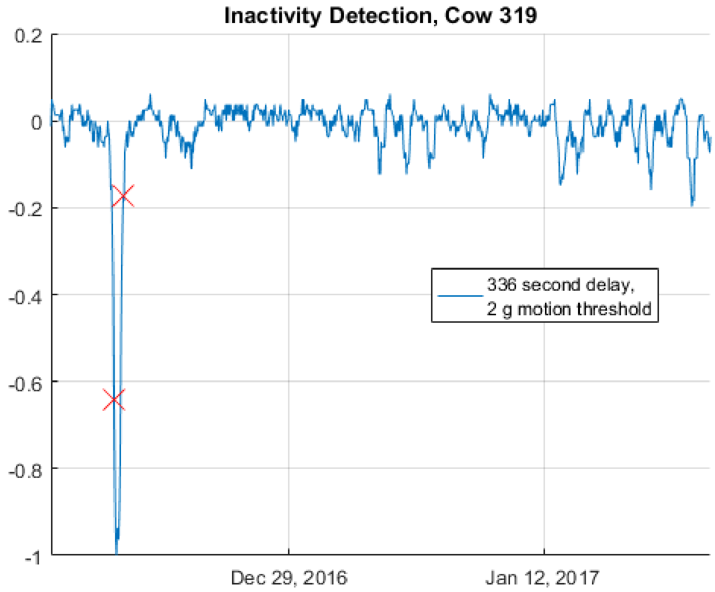

Figure E1: The results of the inactivity detection algorithm of the cow 319. The parameters are 336 second delay and 2 g motion threshold. A true positive estrus is detected on December the 20$^{\text{th}}$. The amplitude of proestrus and non-estrus is significant.

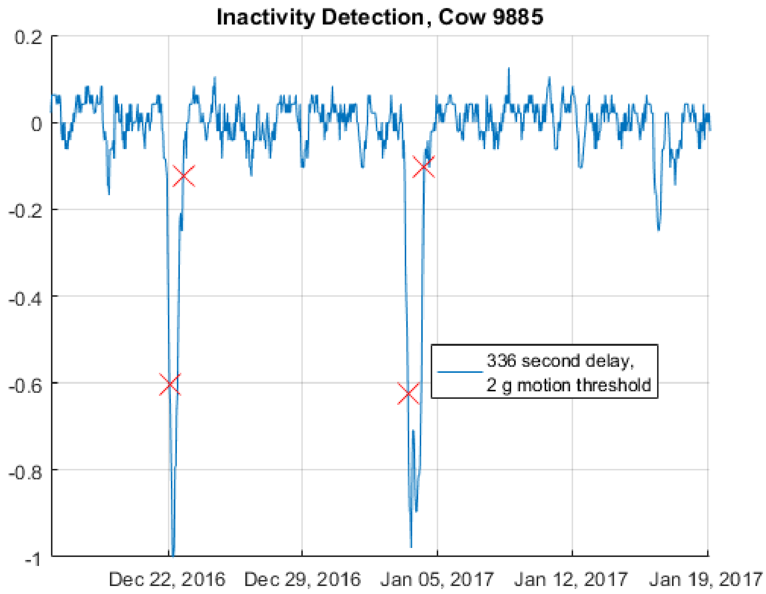

Figure E2: The results of inactivity detection algorithm of the cow 9885. The parameters are 336 second delay and 2 g motion threshold. Both of the estruses are detected as true positive on December the 22nd and January the 4th. No false positive or false negative detection occurred. The amplitude difference between proestrus and non-estrus is obvious.

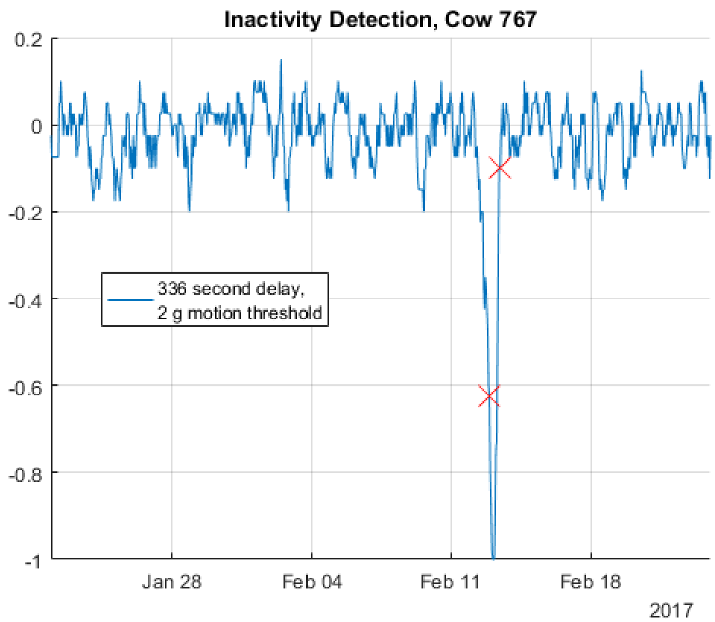

Figure E3: The results of the inactivity detection algorithm of the cow 767. The parameters are 336 second delay and 2 g motion threshold. A true positive estrus is detected on February the 13th. Additionally, the amplitude of the proestrus differs from none-estrus significantly.

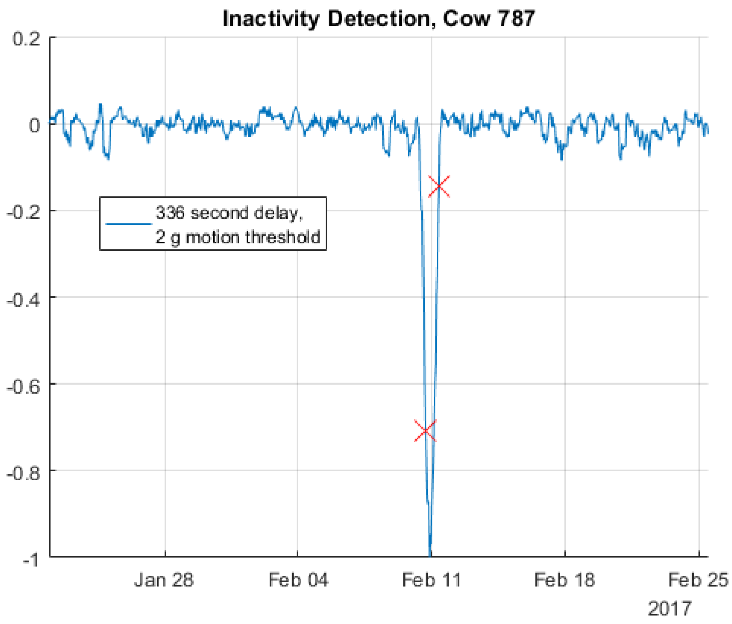

Figure E4: The results of inactivity detection algorithm of the cow 787. The parameters are 336 second delay and 2 g motion threshold. A ture positive estrus is detected on February the 11th and no false positives of false negatives occurred. Furthermore, the difference between estrus and other periods is most significant.

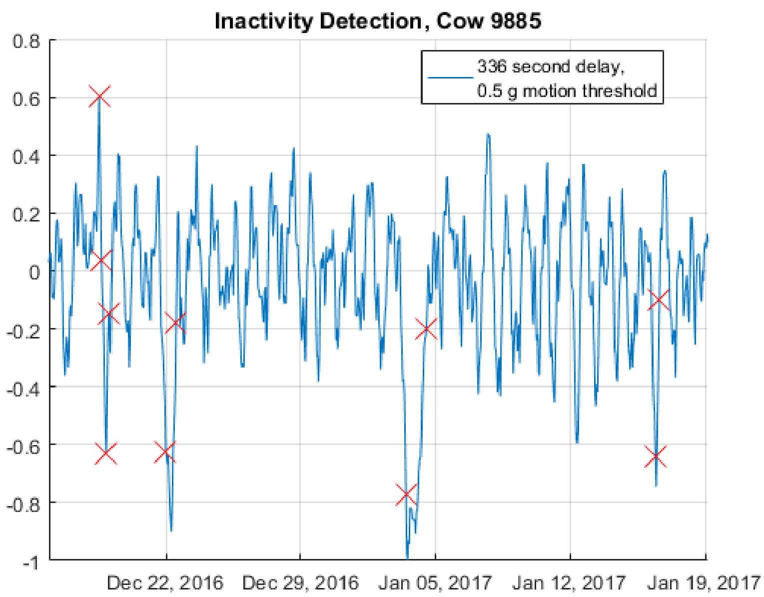

Figure E5:

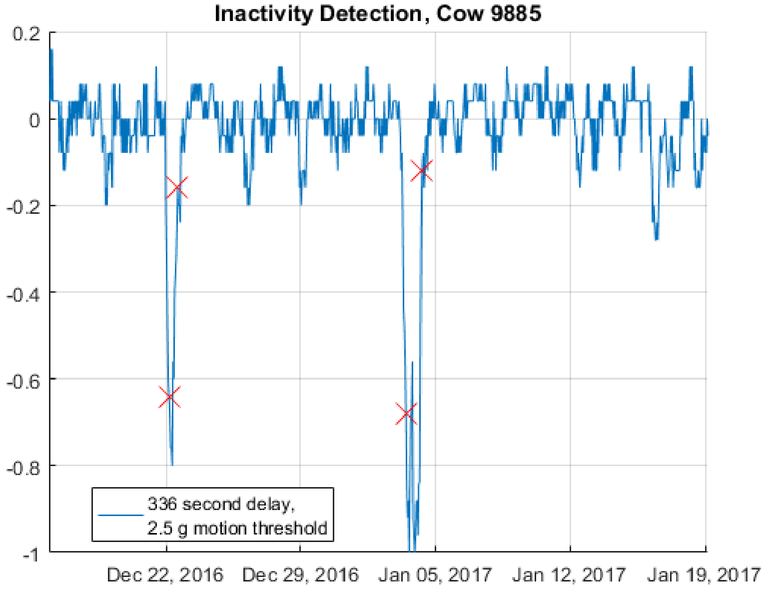

Figure E6:

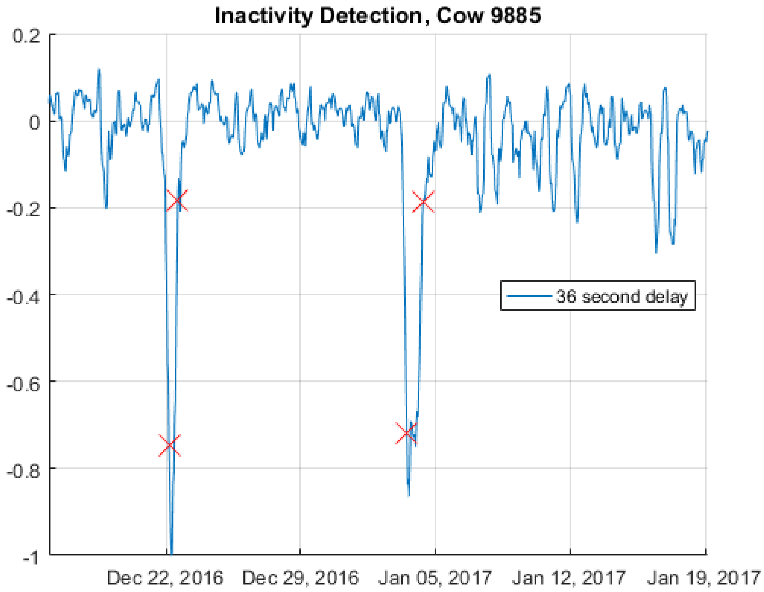

Figure E7:

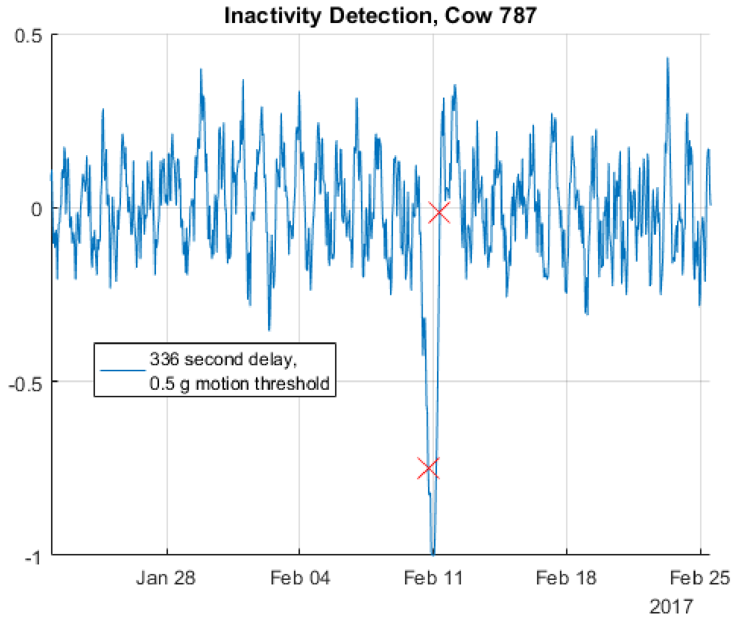

Figure E8:

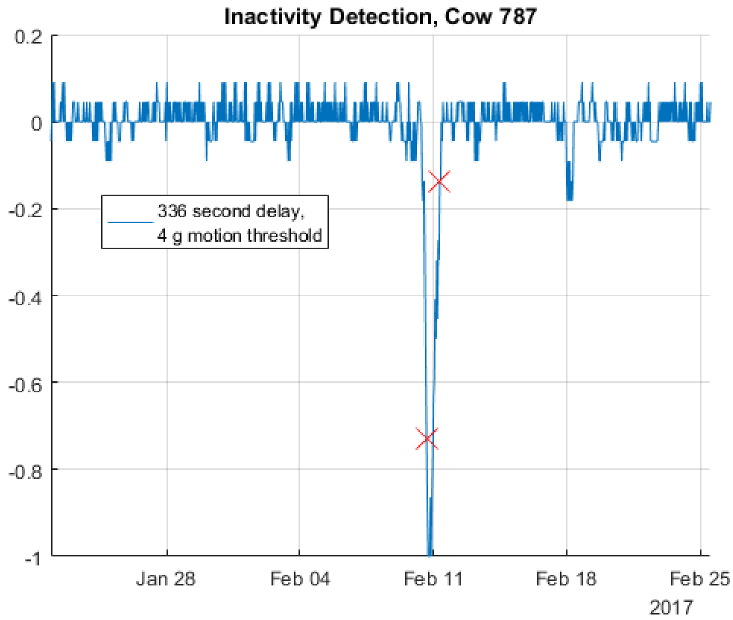

Figure E9:

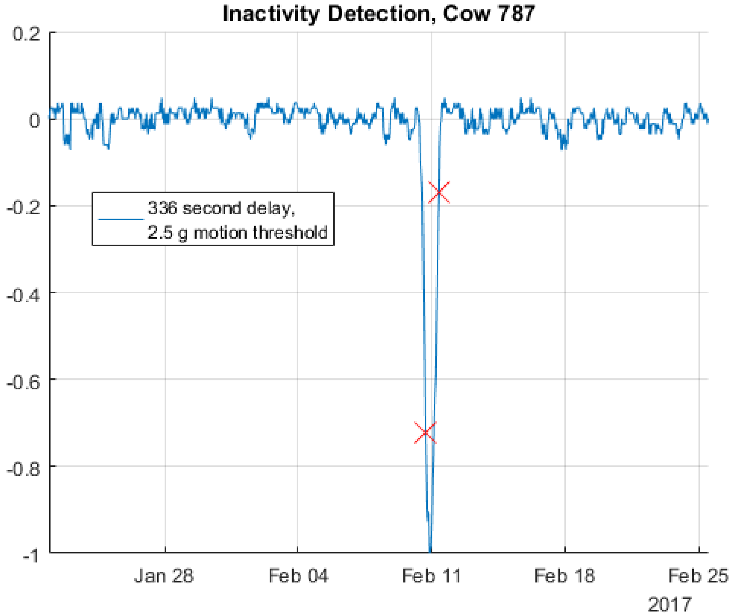

Figure E10:

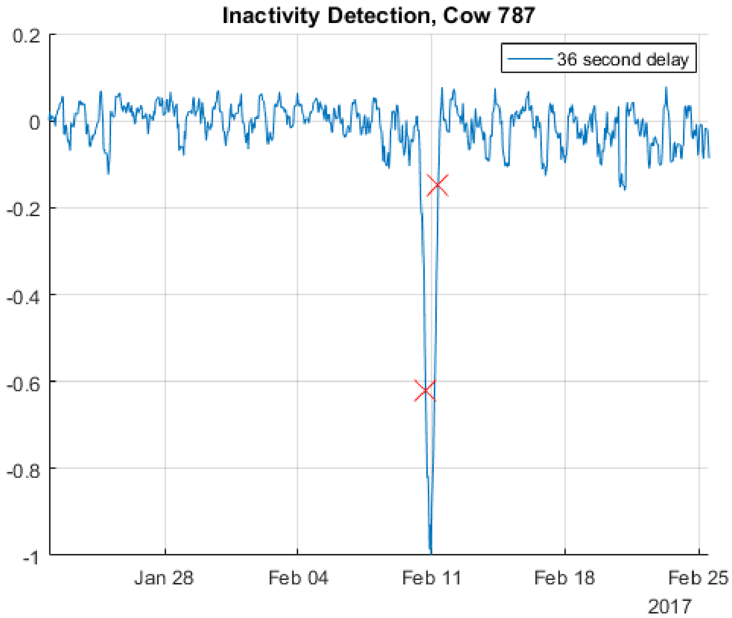

Figure E11:

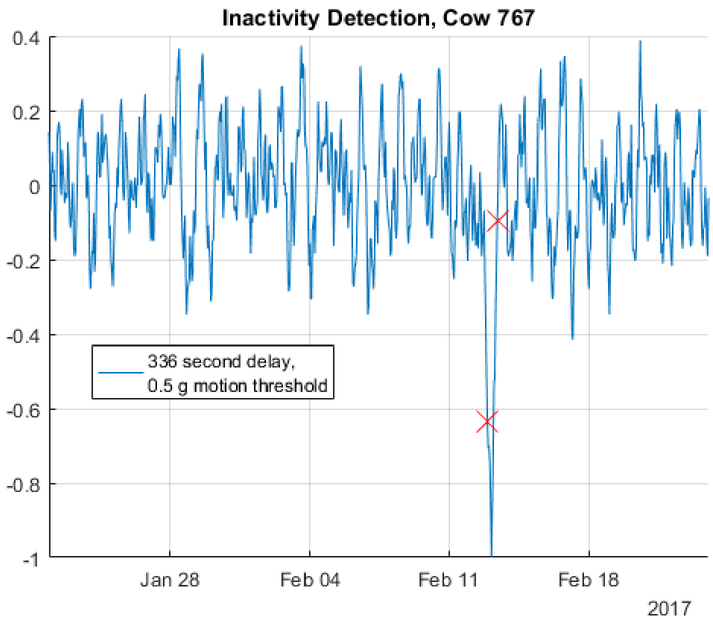

Figure E12:

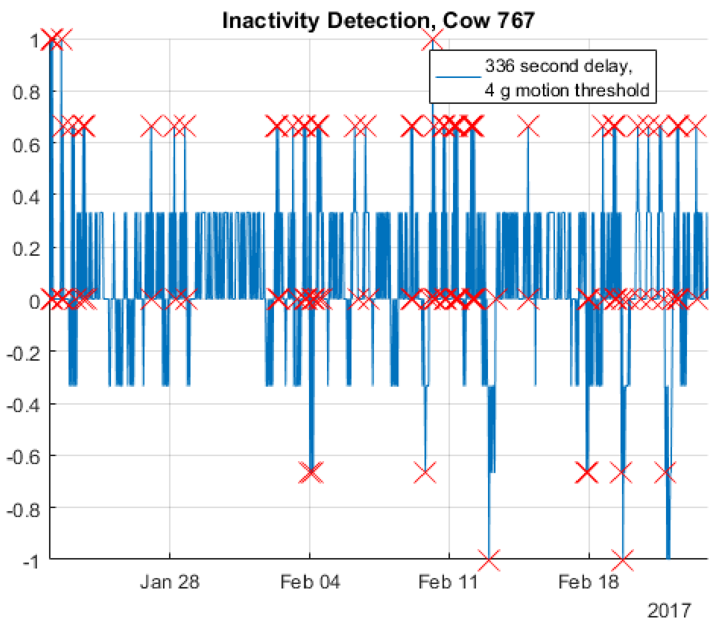

Figure E13:

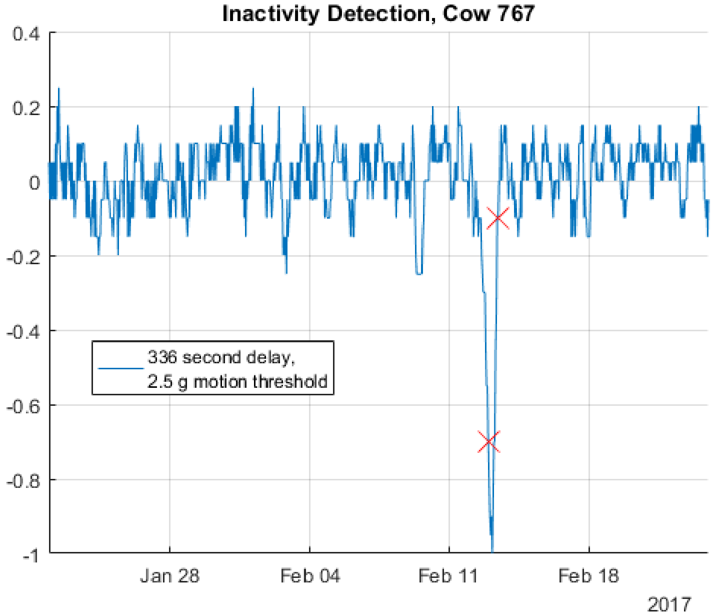

Figure E14:

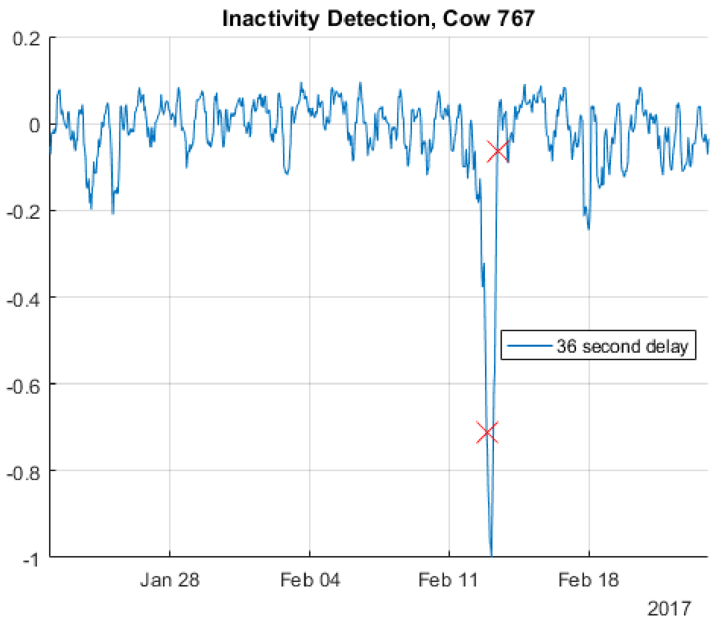

Figure E15:

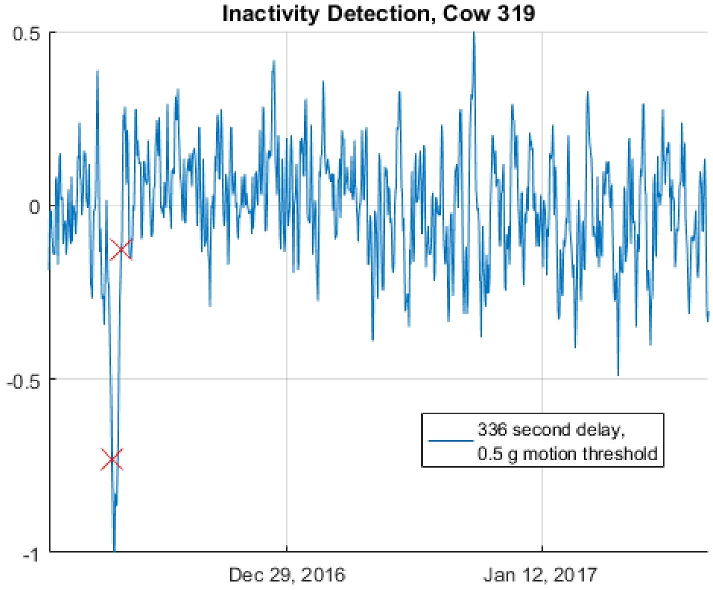

Figure E16:

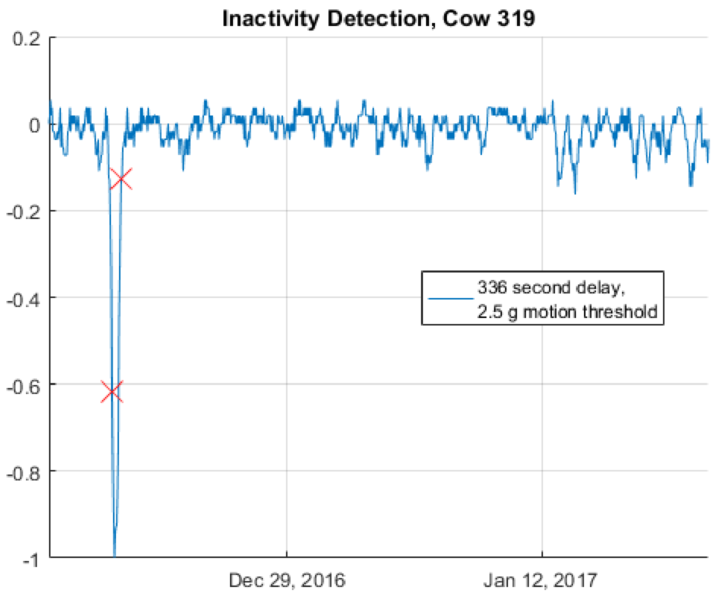

Figure E17:

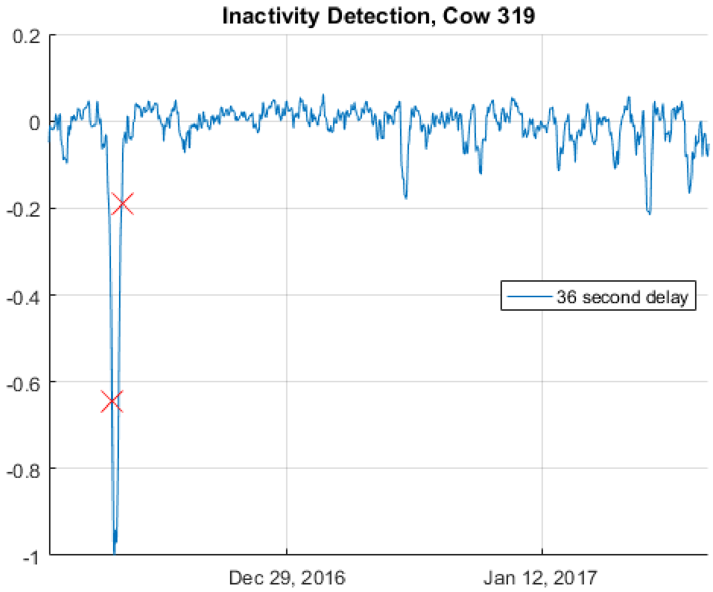
Inactivity Detection, Cow 319
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
36 second delay
Dec 29, 2016
Jan 12, 2017

Figure E18: