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+61
Department of Automations and Systems Technology		
Professuuri: Jotakin kivaa		
Työn valvoja: Prof. Arto Visala		
Työn ohjaaja: M.Sc. (tech.) Samuli Mäkinen		

Tiivistelmässä on...

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+61

Department of Radio Science and Technology

Professorship: Circuit theory

Supervisor: Prof. Arto Visala

Advisor: M.Sc. Samuli Mäkinen

Your abstract in English. Try to keep the abstract short, approximately 100 words should be enough. Abstract explains your research topic, the methods you have used, and the results you obtained.

Keywords: Resistor, Resistance,
Temperature

Prologue

I would like to thank you everyone. [35] [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
$\frac{\mathrm{d}}{\mathrm{d}t}$	derivaatta muuttujan t suhteen
$\frac{\partial}{\partial t}$	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

In master's thesis the introduction section should cover from 2–4 pages. No subsections required. The introduction should explain the following:

Background of the research. Read books and interview persons for this.

Research problem might be difficult to define. However, it comes clear repeating question "why" instead of questions "what" or "how". Why am I doing this work? Why it is so important? Why is it interesting? Because continuous-time observing without using technology is expensive. It is impossible to observe tens or hundreds of cows without technological equipment. This functionality can be integrated with others in same device.

Research objective? The objective of this research is to provide alternative methods for observing health of cows. This new method should overcome older methods in cost, usability and so forth.

Restrictions of the research. What is included and why? What is excluded and why? This research focuses only on cows of milk farms. Other cows and animals are excluded in spite of the possibilities of using same technologies but other algorithms. Also the device and hardware are limited. The power consumption and battery saving sets limits for the possible algorithms. The usability sets restrictions for the maximum size of the device.

2 Background

Originally, humans were hunter-gatherers who obtained food by collecting plants and pursuing wild animals. The methods for obtaining food have changed substantially since the beginning of agriculture. Plants and animals are grown centralized by farmers. Moreover, it has been estimated that the animal husbandry has started more than 10,000 years ago in western Asia. Accordingly, goats were among the first domesticated animals in history [34]. Thereafter, people have been domesticating other species e.g. cows, sheep and pigs for food, milk and other animal products. In addition to the number of species, the head count has been growing with human population (lähde?). Furthermore, the industrial revolution has started a trend of increasing farm sizes and a loss of small farms. Considering only cow farming, alone in the United States of America, there were nierly 90 million cows and calves in 2014 [2]. Respectively, the head count world wide has been estimated almost up to 1,000 million heads in 2016 [3]. Nowadays, the cattle rearing is divided into two trends of beef and dairy farming. Moreover, the breeds for beef cattle and dairy cattle are different. The scope of beef breeding is in rapid brawn growth, whereas it is high milk yield with dairy breeding. In despite of the scope of this study in dairy farming, some of the results of this study could be adoptable and beneficial also in beef farming. Eventually, it is very likely for both of the breeds to end up in food or animal feed.

According to the title, the scope of this study is in dairy farming and specially in estrus detection. Thus, it is mandatory to explain the principles of dairy farming and study dairy cow in general. Furthermore, it is necessary to be acquainted with milk yield related lactation end estrus cycles. Otherwise, the comprehension of the reason for the estrus detection will not be unambiguous. Therefore, we start with the basics present-day of dairy farming. Even today the technological progress in dairy farming is dispersed. Thus, we need to discuss of both, conventional and modern dairy farming. Next, we take a look on a generalized dairy cow. Moreover, we discuss of its natural and farm environments. Furthermore, we discuss in detail of the lactation and the estrus cycles and how they affects on milk yield. Milk yield affects directly to farm profitability. Consequently, it is the fundamental motive for this study. Second lastly, we study currently utilized health monitoring and estrus detection methods and technical aids. Furthermore, we discuss of their assets and disadvantages. Lastly in this section, we discuss of most resent research and development on health monitoring and estrus detection devices.

2.1 Dairy Farming

As discussed in previous subsection, the origins of animal husbandry are over 10,000 years old, whereas, the drinking of milk started 8,000 years ago in Turkey area. Thousand years later, dairy farming started to spread to Europe and next to Africa 6,000 years ago [34]. Thereafter, dairy farming has spread all over the world and 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 (lähde). Nowadays, dairy products have an essential part in human nutrition. Milk and dairy products are produced more than never before. The farm sizes have been increasing meanwhile the number of farms has started to decrease. In 2015 in Finland there were total of 909,000 cows of which 282,000 milking cows[20]. In comparison in 2014 in the Unites States of America, there were more than 89 million cows and calves of which more than 9 million were milking cows[2]. In the end of the year 2015 in Finland, 7890 farms delivered milk to milk processing plants. The average yield of the farms was 279 thousand liters and the average yield of a cow was 8300 liters. In summary the total yield in Finland was 2365 million liters. [20] In the USA the total farm income in cash in the year 2014 was over 49,349 million dollars [2].

Originally, cows were wild pasture animals domesticated by humans. Moreover, people were migrating nomads. Thus, in the beginning of animal husbandry the cattle traveled with people. Finally, people started to settle in constant regions with the animals. First, animals were held in yards but people started to build structures for keeping and protecting the animals. Next, people started to keep cows in buildings. In linear pottery culture, people and animals lived together inside longhouses. Finally, people started to build separate building called cowshed for keeping cattle. Currently, the cowsheds are divided in two types of tie-stall and loose-house cowsheds [16]. In general, tie-stall cowsheds are smaller and 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. Additionally, they may have free access to pasture. Recently, the tie-stall cowsheds have been under critique. That is, the cows are not able to behave as social animals. Moreover, monitoring and keeping health is more difficult in tie-stalls. Therefore, new-builds are rather loose-housed than tie-stall cowsheds. In addition, it is easier to monitor the health and estrus behavior of freely moving animals. Thus, cows living in a loose-housed cowshed are in the scope of this study.

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, 16].

Typically, cows lay down approximately from $11\,\mathrm{h}$ to $12\,\mathrm{h}$ every day day. Meanwhile, they stand up and change their pose several times [38]. 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 [26]. On pasture, their daily walking range may extend to several kilometers [26, 13]. However, cows are very cautious animals. Thus,

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.

Figure 2: Cows in a loose-housed-cowshed [27]. Cows are able to move freely and act as social animals.

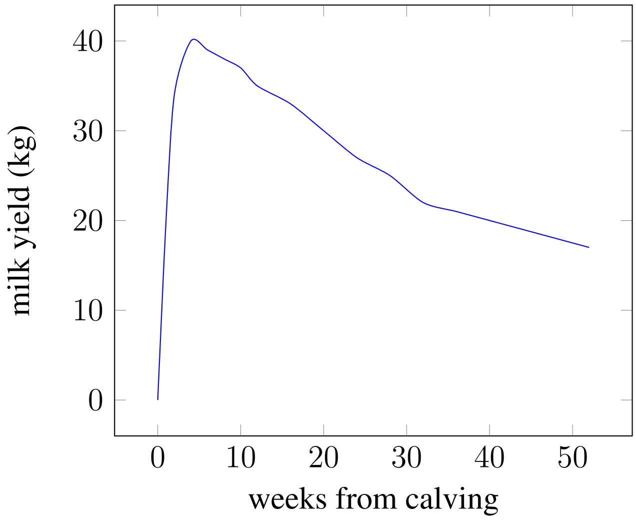

Figure 3: the lactation curve [21]

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 affects to their health and consequently to profitability [16]. Nevertheless, Walking enforces their health, increases hormonal activity and metabolism [26].

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 **??**.

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. [26]

2.1.2 Lactation Cycle

In previous section we discussed of dairy farming and dairy cow in very general level. Conversely, this section is in the core of the scope of this study. That is, lactation cycle is strictly related to the milk yield and, thus to profitability. Furthermore,

Lehmien tuotos kasvaa jokaisella poikimiskerralla [1].

Laktaatio kierrosta, perusasiat, kesto, merkitys maidontuotannolle yms.

2.1.3 Estrus Cycle

Kiimakierrosta, kesto, vaiheet, mikä merkitys jne.

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]

Estrus a.k.a standing heat.

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 detecting estrus and health issues. 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

In this subsection we will discuss of currently used methods for observing the cattle. The discussion starts from conventional and non-technical methods and ends to current commercial technical products.

Traditionally, health monitoring and estrus detection have been based on purely visual observations. However, this method is not considered efficient. In the USA, 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]

Moreover, the only method for monitoring and detection has been observing the cows sight-wise [16]. 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. That is, the larger the cattle, the more work and, thus, the less time available. Consequently, the available time per cow decreases exponentially when the amount of cattle increases. In addition to the experience level of the tender, it is fundamental to know the cattle.

In despite of the challenges, observations of cattle tender are still common monitoring method in small farms and in developing countries. [12]

Therefore, the cattle tender must know his cattle well in order to receive satisfying results. The larger the cattle

2.2.2 Research and Development

Former subsection discussed of currently used farming technologies. In contrast, this subsection reveals technologies under research and development and discusses future options.

This subsection reviews the wearable sensor devices used in dairy farming. The review includes currently used technologies as well as technologies that have been under research but have no been commercialized yet.

behavior pattern recognition using three-dimensional accelerometer and support vector machines

Used three-axis accelerometer and support vector machines for detecting the following behavior of cows:

- Standing
- Lying
- Ruminating
- Feeding
- Normal walking
- Lame walking
- Lying down
- Standing up

The sampling frequency was $10\,\mathrm{Hz}$ and the data was sent to a PC computer via $2.45\,\mathrm{GHz}$ radio band. Measurement range of $\pm$ 3 g. The sensor was ADXL330, Analog Devices Inc., USA and it was attached to the neck. They used 30 Ayrshire and Holstein-Friesian cows. The cows were loose-housed and the barn used automatic milking system. Most of the days the cows had a free access to a pasture. The data was recorded during 30 days. The video, stopwatches and accelerometers were synchronized manually. Cows had different gait scores . . . Used multi-class support vector machines. $70\,\%$ of the data was used as training set and $30\,\%$ was used as a test data set. Lying head erect does not differ much from standing posture considering the accelerometer data. This causes misclasifications. The interest war predicting the behaviour that occurred instead of the behaviour that did not occur. Fixed parameters (e.g. time window) are not suitable for all different behaviours. [22]

Six dairy cows monitored continuously for 36 h. They used a decision-tree algorithm for detecting if the cow was either standing, feeding or lying. The algorithm matches the performance of computationally more intensive algorithms such as hidden Markov model and support vector machines. It is suggested that the decision-tree algorithm could be a part of a real-time behavioural monitoring system. The performance of the algorithm varies in function of the time window which was from 1 to 10 minutes. In the algorithm comparison table the decision-tree algorithm was better in 1 minute window than in 10 minute window. The support vector machine, however, was equally good or even better algorithm. Holstein dairy cattle located in Essex, UK. The cows were loose-housed in a cubicle shed. The herd was milked three times a day. The cows selected for the study did not show signs of lameness or other deseases. Drinking, brushing and walking activities were excluded in this study. [37]

Combines step count and leg tilt data. Data was extracted from 44 dairy cows over an 8 month period. The results show sensitivity $88.9\,\%$ and error rate $5.9\,\%$. The sensor was IceTag3D ®which is a three-axis accelerometer. The sample period was chosen to be 1 minute and is the time resolution through the study. The sensor provides four variables: the percentage spent in three of the stages and step count. The stages are lying, standing and motion. The lying and standing are detected by the tilt of the sensor. The variables lying and standing gives the percentage of the sampling period spent in each of the two states; the motion index is a measure of how much the cow has moved in the sampling period and step count is the number of steps in each sampling period. The measurements were available for a total of 88 cows over varying periods. The data transfer from the IceTag was done manually holding a reader close to the IceTag device. Data sequence for estrus detection were selected around 18 and 23 days between two estruses. 62 cows did not get pregnant, 18 cows were inseminated wearing an IceTag and 44 were not inseminated at all wearing an IceTag. Total of 26 cows became pregnant. The conclusion says, this method could only improve other estrus detecting methods but not alone reliably detect estruses. [17]

They used pedometers. The data set consists of data gathered from 98 cows successfully inseminated by visual estrus detection. The pregnancy was considered as a confirmation for the detection. In the data gathering period of six months a total of 335 estrus cases occurred. 237 cases were discarded since the cows either did not become pregnant or did not wear pedometers. They could improve the visual estrus detection up to 84.2 % accuracy. [12]

This research aims to provide a cheap sensor for estrus detection which is an alternative for current commercial products basesd on accelerometers and odometers. The provided sensor is a wireless intravaginal probe measuring temperature and conductivity. The benefit of this probe against accelerometers is that the cow is not required to move for estrus detection. In the study a video camera was used for detecting the action of the cow, since, moving, standing, eating and stress increases the body temperature. It is suggested that an accelerometer could be used instead of videocamera for automated motion detection. The conductivity measurements were not satisfying since, the contact between the animal and the tissue vere not constant and caused noisy measurments. Yet, some correlation were detected with the cow activity. [5]

in the further study the probe measured vaginal temperature, acceleration and vaginal tissue resistance. They had two kinds of probes, a 160 mm $160\,\mathrm{mm}$ probe and a 1 $120\,\mathrm{mm}$ probe. The longer one caused bleeding during the trials. The shorter one worked out fine but started to rotate during the estrus and it even ejected. Therefore, no sufficient probe desing was found. The 3-axis accelerometer provided different spaces for datapoints when the cow was either standing up, resting wide or resting narrow. [4]

Four wireless accelerometers were used per cow, one on each limb. In the research the wavelets were analyzed and mismatch in symmetry of variances observed for lameness detection. Sampling frequency was $25\,\mathrm{Hz}$ 25 Hz. [25]

3 Research

In previous section 2, the backgrounds of this study were in discussion. The discussions covered the dairy farming in general level and the dairy cow more in detail. Moreover, the section focused in milk yield related estrus and lactation cycles. Additionally, we discussed of current and future solutions for dairy cow health monitoring and estrus detection. In conclusion of the first section, there is a certain need for efficient and cost-effective solutions for dairy cow monitoring. Accordingly, wearable wireless sensor devices are currently promising due their overall performance and availability. Respectively, in this section the target is to develop convenient estrus detection algorithm for a wearable wireless sensor device. Furthermore, the development initiates from a scratch. That is, no assumptions are made based on previous studies.

In this study, the research section has been divided in three subsections. The first subsections 3.1 discusses of data recording. The discussion includes the description of the hardware as well as the software implementation. The hardware description covers the main hardware components in reasonable detail. Additionally, we discuss of the communication between the main components. Respectively, the software implementation description discusses of the work flow of the data recording software.

The second subsection 3.2 introduces a set of methods applied on the recorded data. The set consists of basic statistics as well as various digital signal processing methods. The basic statistics contains only the most fundamental measures of statistical analysis. Nevertheless, they are discussed only briefly in this section. Conversely, the digital signal processing methods contains more advantageous calculus. Thus, they are discussed more in detail.

Lastly in this section 3.3, we develop three different algorithms for dairy cow estrus detection. All these algorithms have their basis in data processing methods introduced in previous subsection 3.2. However, each of them have slightly different approach to detect the estrus. Nevertheless, all the algorithms detects rather the pro-estrus than the actual estrus 2.1.3. All these differences are covered in the discussions. Furthermore, we discuss of the background and describe the evolving of each algorithm in detail.

Tähän ehkä jotain hypoteesia tutkimuksesta ja jotain mahdollisia ennakko-oletuksia?

3.1 Data Recording

Formerly, we described the research of this study in general whereas, the topic of this subsection is in the data recording. Altogether, the data recording is divided in three more sections. The first describes the data recording hardware. The hardware was a prototype of a sensor device for dairy cow monitoring. Furthermore, the hardware was modified for data recording purposes. Nevertheless, any hardware design is not included in this study. Next, we discuss of two different software implementations for data recording hardware. The starting point for the first implementation is making of no assumptions whereas, the second implementation is based on the results of the results of first recorded set of data. Lastly in this subsection, we discuss of the data recording process itself.

3.1.1 Hardware

As discussed previously, the hardware for the data recording was a prototype of a dairy cow sensor device. Originally, the hardware consisted of micro-controller unit (MCU), accelerometer and thermometer. However, it lacked of storage memory for storing the recorded data. Thus, the hardware was customized and a Secure Digital (SD) memory card slot was added. Otherwise, the hardware remained the same during the entire study. In despite of the customization, the hardware design is not in scope of this study. Therefore, the following is rather description of the hardware than comprehensive discussion of the design itself. In addition to hardware components, we briefly discuss the communications between the components. In general, the main hardware components brief descriptions are as follows:

- *Atmel ATmega32u4* is a high perforamance low power 8-bit micrto-controller. It is designed for optimizing power consumption versus processing speed. It contains 135 powerful Reduced Instruction Set Computer (RISC) architecture instructions of which most executable in 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 use of micro-controller. It is capable of measuring acceleration in three perpendicular axes. 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 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. [35]
- *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 $2\,\mathrm{GB}$. However, High Capacity SD (SDHC) extends the maximum capacity to $32\,\mathrm{GB}$ and Extended Capacity SD (SDXC) up to $2\,\mathrm{TB}$. [30].

Naturally, all of these hardware components are assigned to specific tasks with respect to their functional description. Their special assignments are discussed in detail in the following descriptive sections.

In addition to the hardware components, the hardware configurations consists of two serial interfaces as follows:

- *Inter-Inegrated Circuit (I^2C) bus* or some times referred as *Two-Wire Interface (TWI)* is a serial communication interface developed by Philips Semiconductor. The first

I^2C was released in 1982. The design isn rather simple hence, it requires only only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors [36].

- *Serial peripheral interface (SPI)* [32] kjlk l

Micro-controller

The micro-controller unit (MCU) is the fundamental component of the sensors device in this study. It is responsible of execution of the software flow whenever the device is powered. The software flow starts from the boot, that is, after switching the power on or resetting the micro-controller. During the boot, it initializes itself as well as the connected sensors according to the prior set configurations in the software. After the boot, the micro-controller begins the execution of the repetitive software loop including various tasks and events. The initialization configurations and the tasks and events are discussed more the software section 3.1.2.

As stated earlier, the micro-controller unit of the sensor device in this study is Atmel ATmega32u4. The following describes the main features of the MCU considering this study.

- $32\,\mathrm{kB}$ *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 of the controller. Thus, it enables uploading the application software without external USB module. Additionally, 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, certain input pins can be configured as interrupt pins. These configurations and features are required for such finctionalities as serial communication, setting and getting statuses.

- *Interrupts* are occasions that usually sets an interrupt flag. The Interrupt Service Routines (ISR) corresponding the interrupt flags are executed with respect to their priority. There are internal and external interrupts. The interrupt can be enabled individually. Interrupts are beneficial in generating events into software flow instead of executing tasks periodically.

- *Watchdog Timer (WDT)* of ATmega32U4 counts time on separate on-chip $128\,\mathrm{kHz}$ oscillator. The WDT is capable of interrupting and/or resetting the system. Additionally, it is enabled in most of the power modes.

- *Power Management and Sleep Modes* allows the user to tailor the power consumption to the application's requirements. This es beneficial specially with battery-powered solutions where regular re-charging is not effortless.

- *SPI and I²C* serial bus interfaces are used for internal communication the micro-controller and the sensors as well as the SD memory card.

Accelerometer

The BMA222E is a triple-axial accelerometer used for measuring change in motion as well as position recognition. The axis of the accelerometer are in right angle to they each other. Thus, it can measure in all direction. Additionally, it contains several on-chip functions for prior and post filtering as well as detecting different conditions. The filters and the detectable conditions are configurable and they are discussed in detail later in this section.

This subsection surveys through the most likely worthwhile features of the accelerometer used in this study. However, all of the features discussed features may not be used in this study but may become sensible in future studies. The overview of the hardware of this study was discussed previously. The specific configurations of the sensor are discussed in subsections 3.1.2 and 3.1.2. The accelerometer used in this study is Bosch Sensortec BMA222e digital, triaxial acceleration sensor. Its key features are

- *On-Chip Interrupt Controller* is capable of generating various interrupt. It yields an opportunity to create device applications even without a micro-controller. Additionally, together with micro-controller there i s no need to continuous sampling by the controller.

- *On-Chip FIFO Register* is capable of storing up to 32 frames. Depending on the configurations, one frames contains measurements from one or three axis. Additionally, a frame includes whether the data is new or already read.

- *Range* for 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 linearly.

- *8-bit Resolution* for both, acceleration and temperature measurements. The measurement range of acceleration is adjustable. Thus, the resolution depends on the range. However, the temperature resolution is fixed to $0.5\,^{\circ}\mathrm{C}$ per least significant bit (LSB).

- *Low Pass Filter* enables removing of high frequency distortions from the measured signals. Thus, no additional low pass filtering is needed. The low pass filter is configurable with preset frequencies from 7.81 Hz to 1000 Hz.

- *Offset Compensation* allows removing offsets from the measurements. At sea level, there is always approximately $1\,\mathrm{g}$ offset present. Specially, in integrative calculations it might accumulate and yield misleading results.

- *SPI and I²C digital interfaces* are necessary interfaces for both, configuration and communication between the sensor and the micro-controller.

- *Low Power Consumption* is beneficial in portable and battery-powered solutions such as the sensor device in this study.

- *On-chip Temperature sensor* of the controller provides resolution of $0.5\,^{\circ}\mathrm{C}$. The accuracy of the BMA222E is relatively low in comparison to the TMP112 temperature sensor. However, it can be used as reference or verification value with the temperature sensor. [11]

Limited capacities of power sources are significant issue with wireless devices. Therefore, the low power consumption all together with other on-chip features of the accelerometer become useful. That is to say, it is reasonable to maximize the usage of the low power sensor and meanwhile minimize the use of more power consuming micro-controller.

Temperature Sensor

Texas Instruments TMP112 is a high-accuracy temperature sensor for various applications from portable devices to industrial controls. 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. Thus, in the case of positive correlation the temperature sensor can confirm the detected estrus.

The temperature sensor used in this study is Texas Instruments TMP112 high-accuracy, low-power, digital sensor. It is designed for replacing NTC/PTC thermistors in high accuracy applications. Its main features are

- *High accuracy* without calibration. Furthermore, the data sheet provides instructions for calibrating the sensor. Without the calibration the accuracy is
  - $0.5\,^{\circ}\mathrm{C}$ in range from $0\,^{\circ}\mathrm{C}$ to $65\,^{\circ}\mathrm{C}$
  - $1.0\,^{\circ}\mathrm{C}$ in range from $-40\,^{\circ}\mathrm{C}$ to $125\,^{\circ}\mathrm{C}$
- *High resolution* of $0.0625\,^{\circ}\mathrm{C}$ in both, 12-bit and 13-bit mode
- *Low power consumption* and two different power modes:
  - $10\,\mu\mathrm{A}$ in active mode
  - $1\,\mu\mathrm{A}$ in shutdown mode
- SMBus$^{\mathrm{TM}}$, Tow-Wire and $\mathrm{I}^2\mathrm{C}$ digital interfaces
- supply voltage range from $1.4\,\mathrm{V}$ to $3.6\,\mathrm{V}$
- *Conversion rate* from $0.5\,\mathrm{Hz}$ to $8\,\mathrm{Hz}$
- *12-bit resolution* from $-55\,^{\circ}\mathrm{C}$ to $128\,^{\circ}\mathrm{C}$. The sensor has an 13-bit mode, when the measurement range is up to $150\,^{\circ}\mathrm{C}$.

[35]

Temperature sensor is soldered on the circuit board, but the skin temperature is conducted from skin to chip via heat conducting aluminum tape.

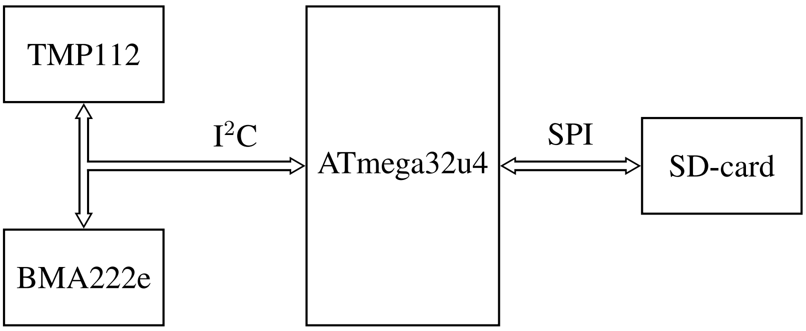

Figure 4: Block diagram of the device. The temperature sensor and acceleration sensor are connected to the micro-controller via I^2C-bus, whereas, SD-card is connected via SPI-bus.

Secure Digital Memory Card

This hardware configuration uses a secure digital (SD) memory card as a memory storage for data recording.

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." [29]

- *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...)." [39]

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.

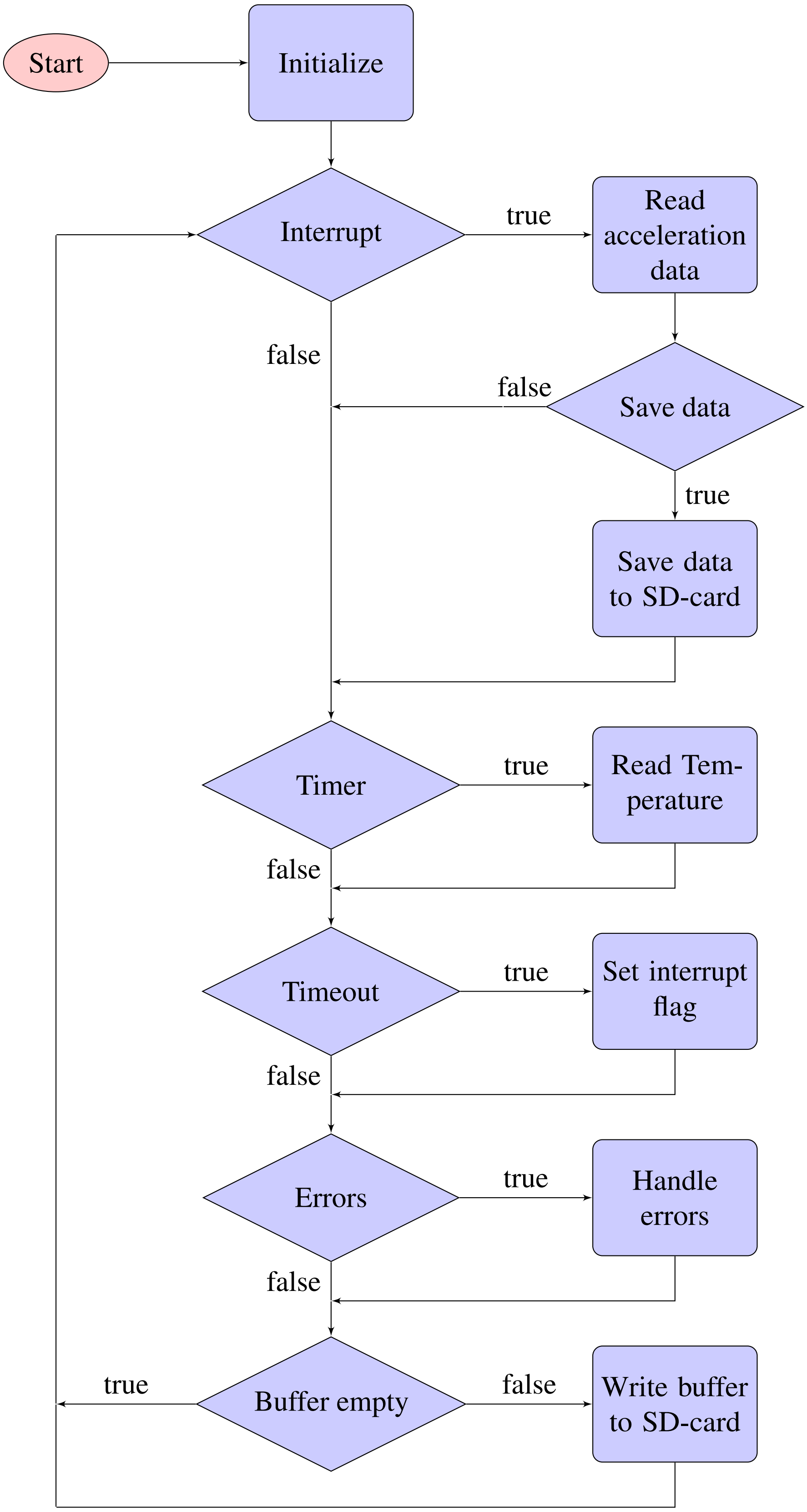

Figure 5: The flow of data logging program

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.
- *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.
- *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

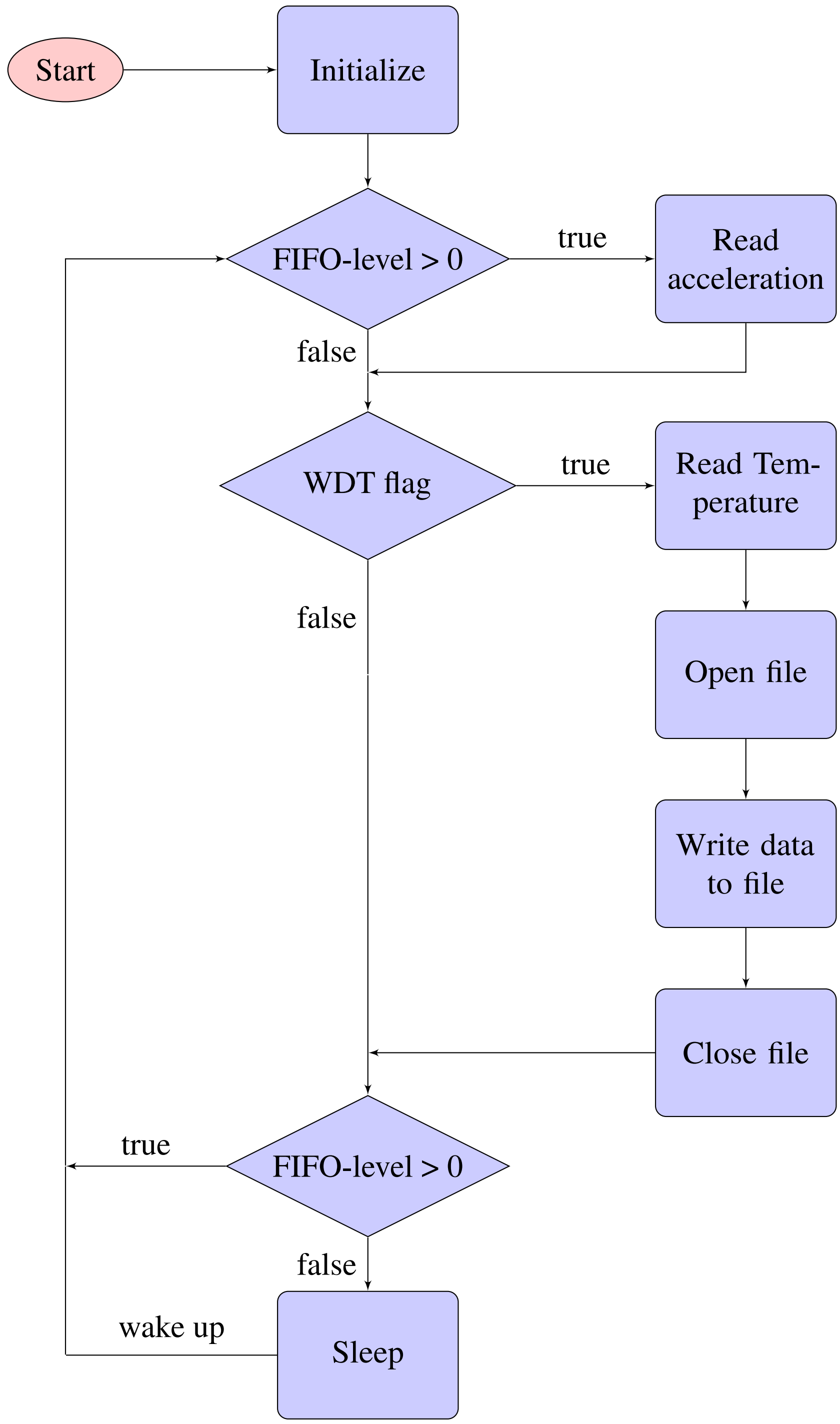

Figure 6: The program flow of the second data logging software

3.1.3 Procedure

All the data used in this study were recorded on a dairy farm in coordinates N$63°6'3.6''$ 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 14$^{\text{th}}$ to 15$^{\text{th}}$. The second phase was recorded instantly after the first and lasted until January the 20$^{\text{th}}$. 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 **??**. Furthermore, statistics are used in analysis of duration of SD file operations in section **??**.

The statistical methods used in this study are defined as follows [31]:

- *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:

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

- *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

Discrete Fourier transform (DFT) is linear mapping of a signal from time domain to frequency domain [28]. 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.

More text here definitely!

3.2.3 Digital Filters

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] [19].

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.

Filters with infinite impulse response (IIR):

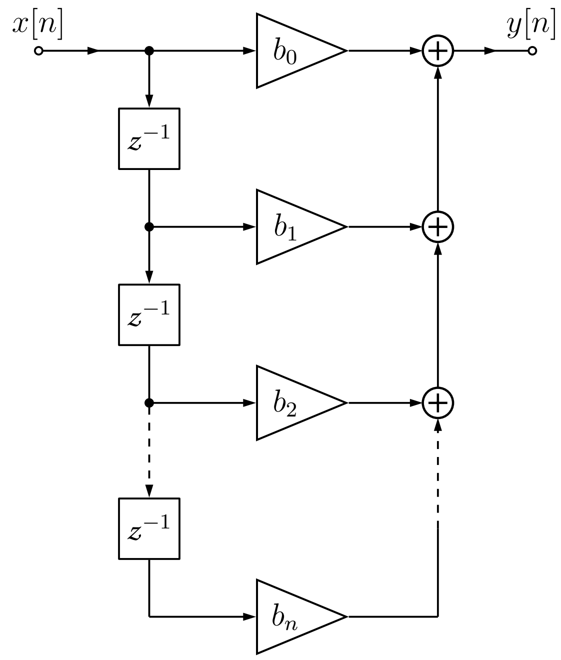

Figure 8: An example of an n^{th} order FIR filter

$$y[k] = b_0x[k] + b_1x[k-1] + b_2x[k-2] + \cdots + b_{n-1}x[k+1-n] + b_nx[k-n] \quad (4)$$

$$\begin{aligned} y[k] = b_0x[k] + b_1x[k-1] + b_2x[k-2] + \cdots + b_nx[k-n] \\ -a_1y[n-1] - a_2y[n-2] - \cdots - a_my[k-m] \end{aligned} \quad (5)$$

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

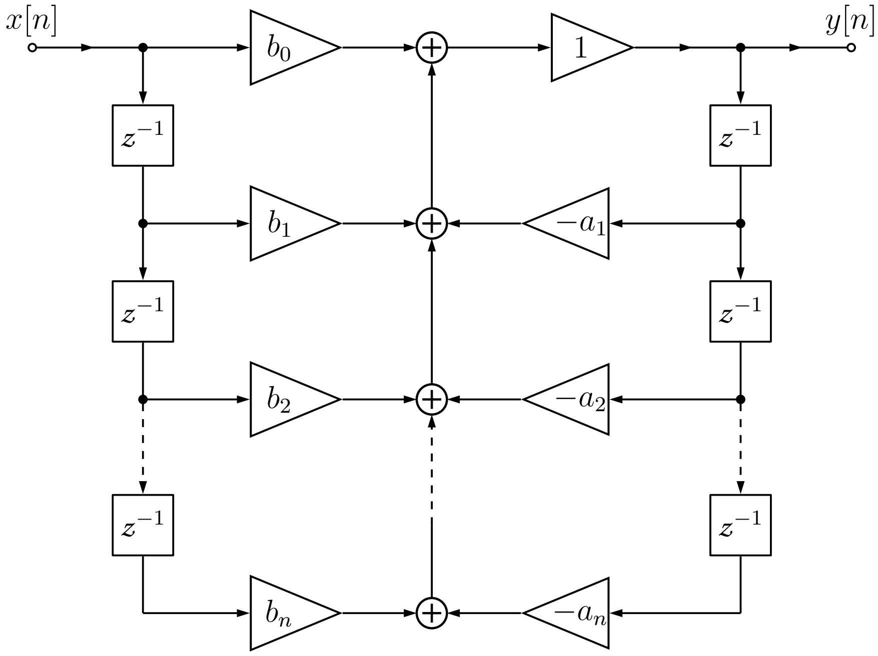

Figure 9: An example of an n-th order IIR filter in direct form I

processing [33, 24] ...

The integration period = window
Windowed mean value:

$$\bar{x}(k) = \frac{\sum\limits_{i=k-n}^{k} x(i)}{n}, \tag{6}$$

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 2.1.3. 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) [23]. 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.

   $$u(k) = \sum_{i=nk-m}^{nk} x(i) \quad \text{, where} \tag{7}$$

   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.

$$u(k) = x(k) - \tilde{x} \quad \text{, where} \tag{8}$$

$\tilde{x}$ 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.

$$u(k) = \frac{x(k)}{\max|x|} \tag{9}$$

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 [18]. 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.

$$y(k) = a_0x(k) + a_1x(k-1) + a_2x(k-2) - b_1y(k-1) - b_2y(k-2) \tag{10}$$

2. **Compute** — The actual core of this algorithm is the length of the acceleration vector and it is defined as
$$u(i) = \sqrt{x^2(i) + y^2(i) + z^2(i)} \tag{11}$$

3. **Sum results**

4. **Remove offset**

5. **Normalize**

6. **Threshold**

7. **Plot data**

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.

2. **Compute Variance** — Compute the variance of the sample

$$u_x(k) = \frac{\sum_{i=nk-m}^{nk} (x(i) - \bar{x})^2}{m} \quad \text{, where} \tag{12}$$

m is the size of the data sample, n is the distance between the samples and k is the index of the output data.

$$u_{tot} = u_x + u_y + u_z \tag{13}$$

3. **Sum results**
4. **Remove offset**
5. **Normalize**
6. **Threshold**
7. **Plot data**

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:

$$u[i] = x[i] - x[i-1] \tag{14}$$

   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**

4. **Remove offset**

5. **Normalize**

6. **Threshold**

7. **Plot data**

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 **??**. 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 10.

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

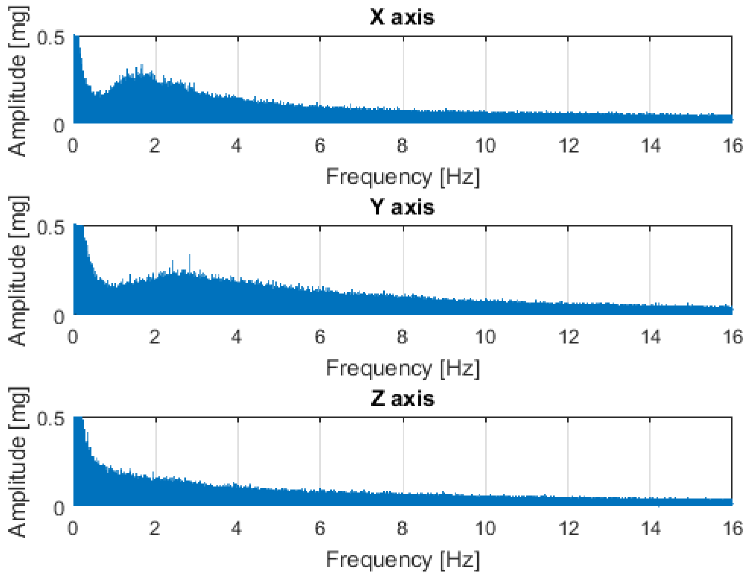

Figure 10: The results of the frequency analysis of the first data set. The frequency spectrum of each axis consists of significant components at frequencies close to zero-frequency and minor components at non-zero frequencies.

Table 1: File operation times of the first data recording hardware and software

Operation	Min [ms]	Max [ms]	Mean [ms]	Variance [ms]
Open file	2	3	4	5
Close file				
Write line				

file size. The minimum opening time was 7.26 ms and the maximum time for 1.37 GB file a was 1440 ms. The average value was 747.14 ms. The file closing times varied from 8.04 ms to 160.03 ms and seemed not to be dependent on the file size. The average file closing time was 19.75 ms. The time spent of writing a single data line was from 56 µs to 162.36 ms and the average was 997.76 ms.

This subsection concentrates in the results of estrus detection. As discussed in section **??** 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.

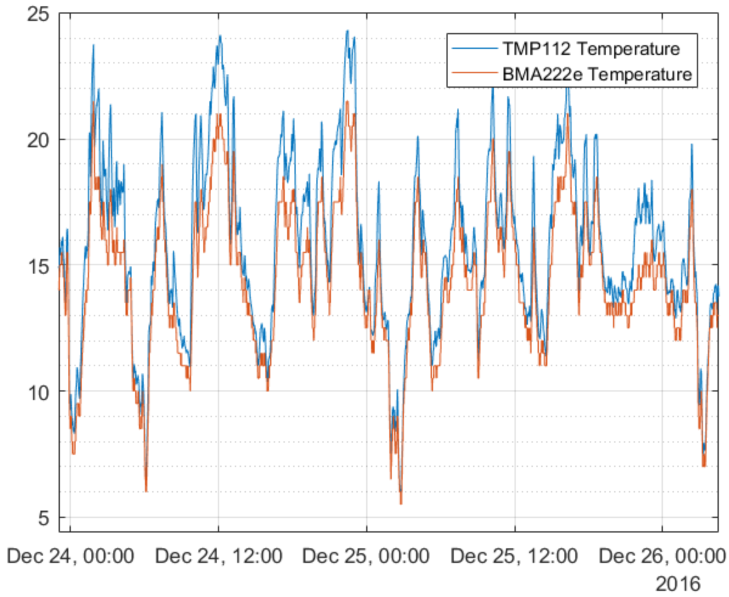

Figure 11: The temperature sensor readings do not correlate with any other results. More likely, temperature correlates with the air temperature inside cowshed.

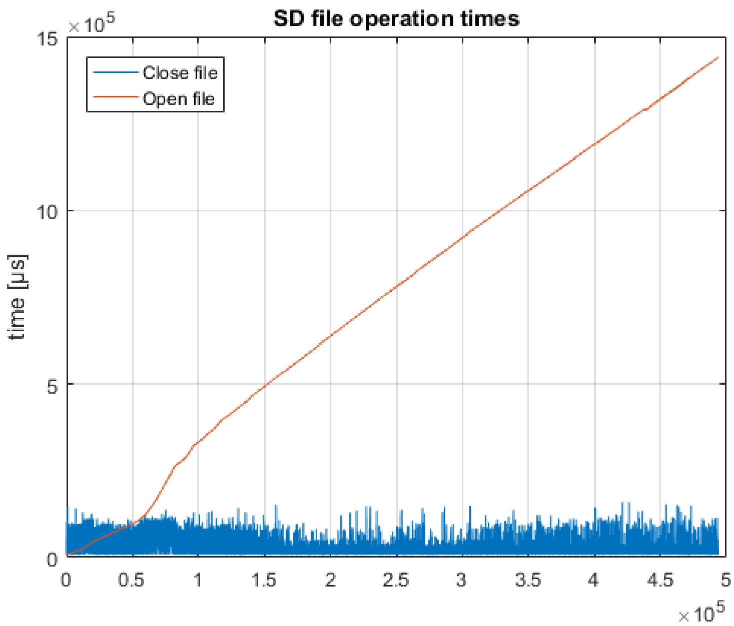

Figure 12:

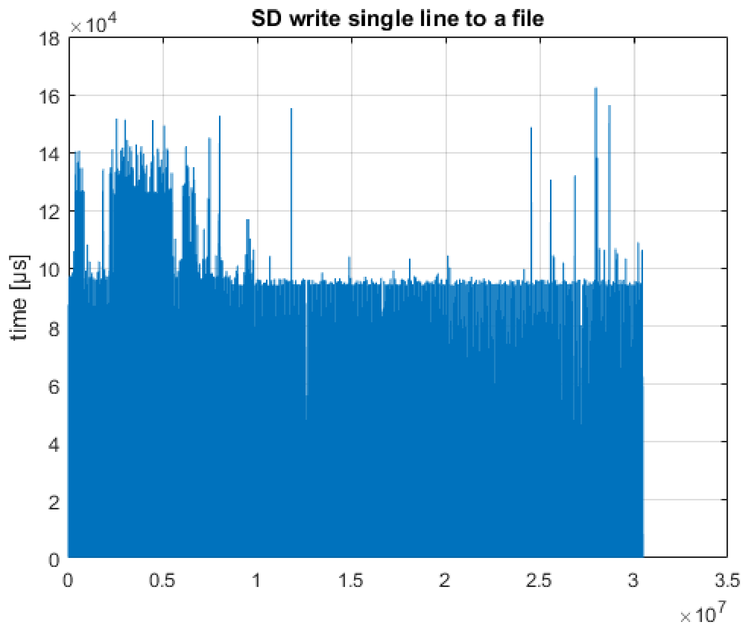

Figure 13:

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 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 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.

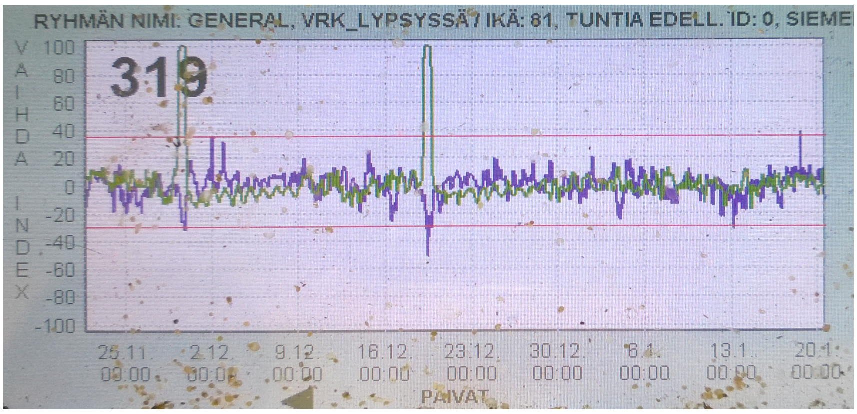

Figure 14: The activity data of Heatime system of the cow 319. The cow has had estruses approximately on October the 30th and December the 20th. The latter of the estruses should be detectable in this study, hence it is within our data recording period.

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

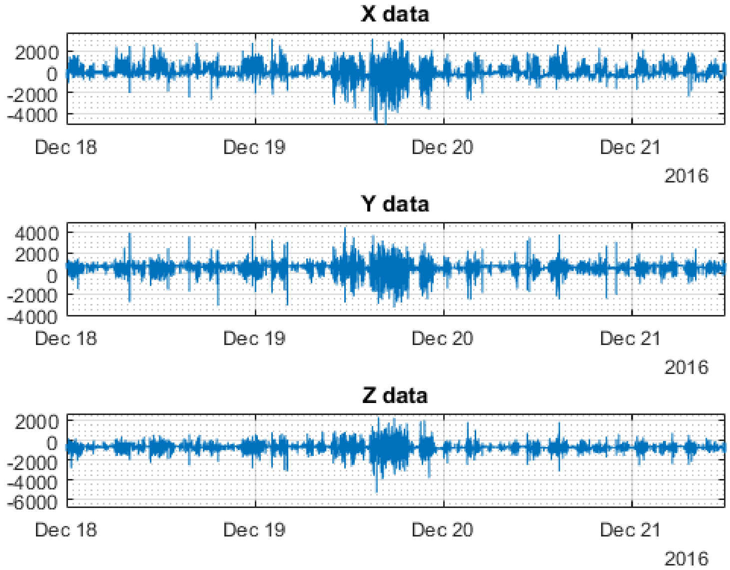

Figure 15: The raw acceleration data of each axis of the cow 319. The time frame of this picture is restricted around the estrus.

long period of high amplitude activity. Furthermore, the change of activity is detectable in all axis of the accelerometer. Figure 15 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.

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 **??**, **??**, **??** and **??** 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.

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 11, 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.

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.

The sensor device did not provide optimal performance for recording large amounts of data. Specially the SD file operations formed a bottle neck in sampling speed.

Oh wow!

4.3.3 Suggestions for Future Studies

More data should be recorded to validate the algorithms. The algorithms should be tunable e.g., the estrus threshold should be individual and controllable.

The results should be verifies by a vet.

The estrus detection should be tested in real-time on a farm

The algorithms should be tested on other breeds as well as cows in tie-stall cowshed but also on pasture.

By successful continuous video recording or monitoring with proper time synchronization it would be possible to detect the activities of a cow.

Behavior detection... rumination lameness...

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

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

[25] 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.

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

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

[28] 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.

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

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

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

[32] *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.

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

[34] 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).

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

[36] *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.

[37] 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.

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

[39] Aaron Wisner. URL: https://github.com/wizard97/ArduinoRingBuffer.

A Heatime Data Plots

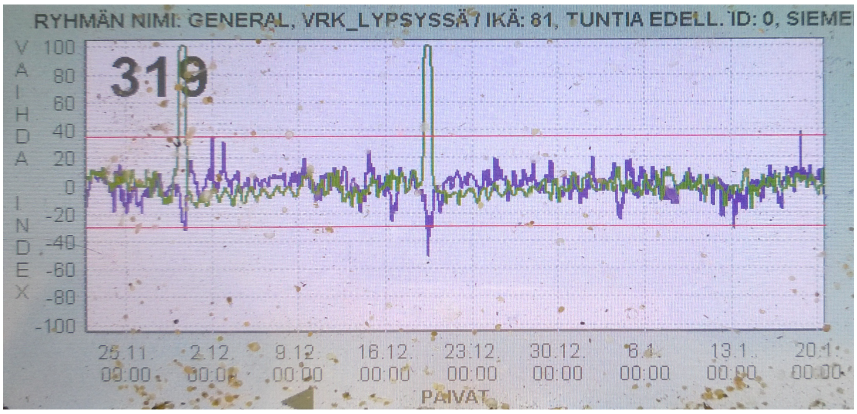

Figure A1: The activity data of Heatime system of the cow 319. The cow has had estruses approximately on October the 30th and December the 20th. The latter of the estruses should be detectable in this study, hence it is within our data recording period.

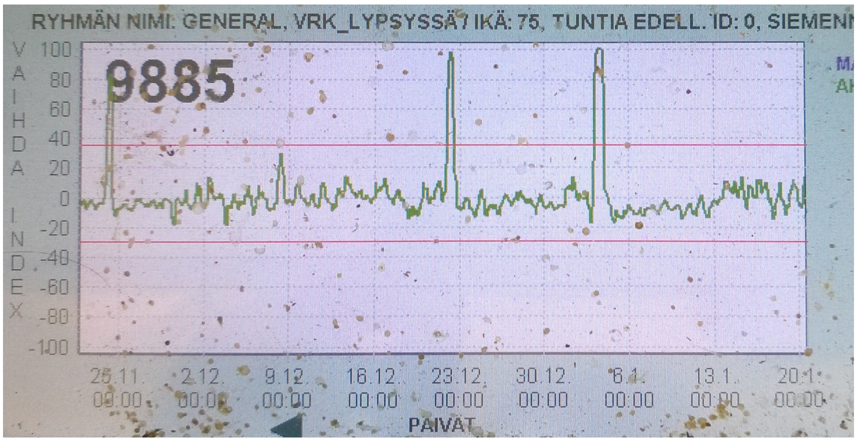

Figure A2: The activity data of Heatime system of the cow 9885. The cow has had estruses approximately on October the 24th, December the 22nd and January the 4th. Two latter estruses should be detectable in this study, hence they both are within our dat recording period.

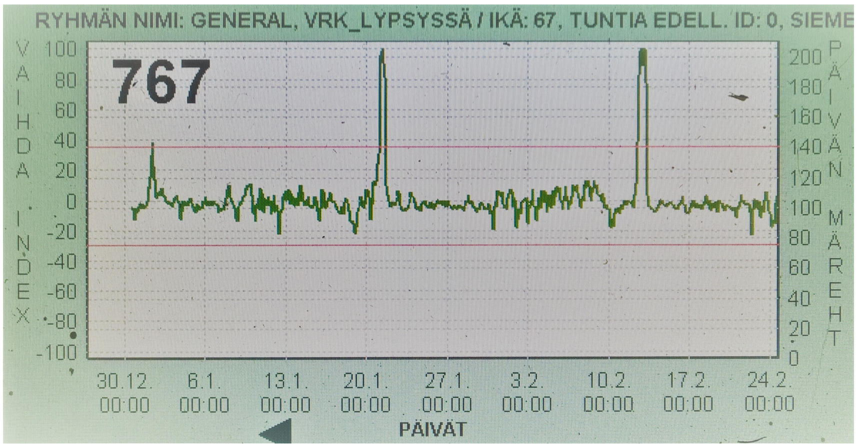

Figure A3: The activity data of Heatime system of the cow 767. The cow has had estruses approximately on January the 21st and February the 13th. The latter of the estruses should be detectable in this study, hence it is within our data recording period.

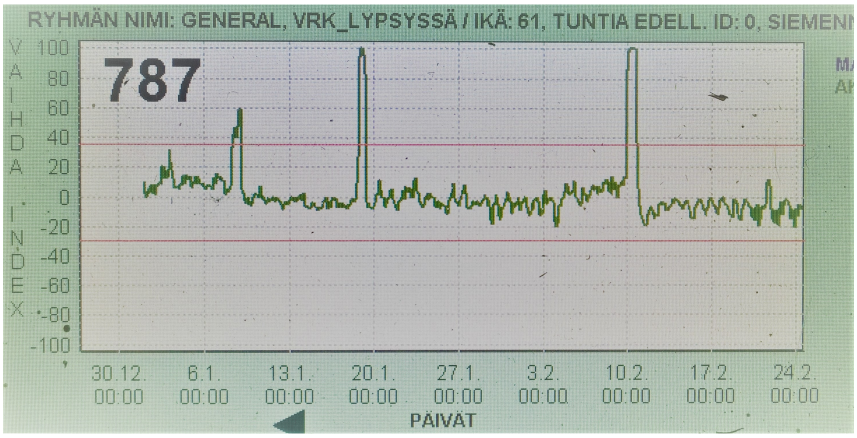

Figure A4: The activity data of Heatime system of the cow 787. The cow has had estruses approximately on January the 19th and February the 10th. The latter of the estruses should detectable in this study, hence it is within our data recording period.

B Raw Accelerometric Data

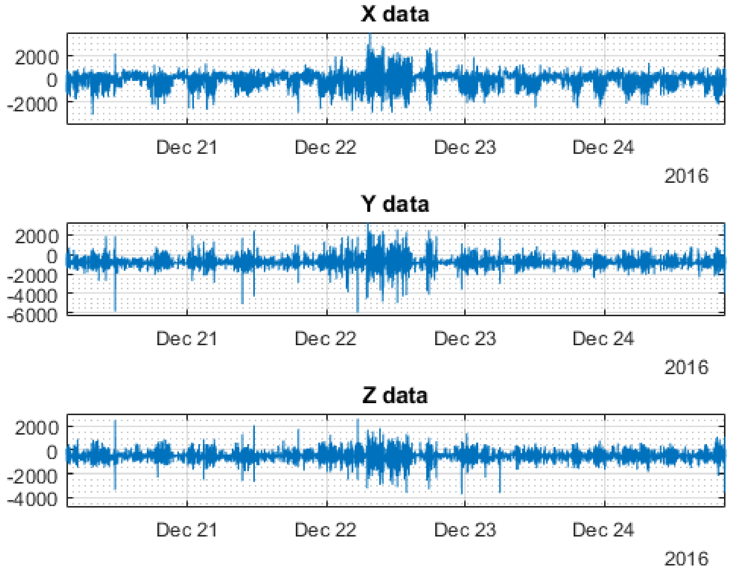

Figure B1: The raw acceleration data of each axis of the cow 9885. The time frame is scaled around the first detectable estrus period.

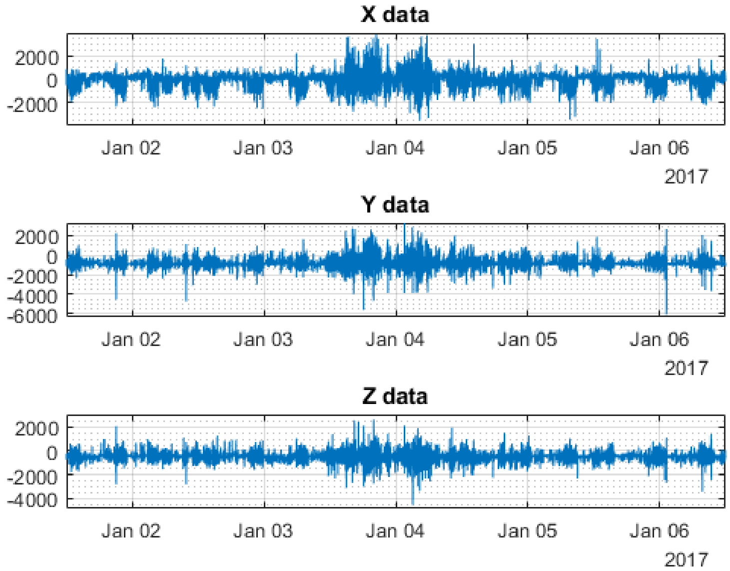

Figure B2: The raw acceleration data of each axis of the cow 9885. The time frame is scaled around the second detectable estrus period.

C Activity Monitoring Full Results

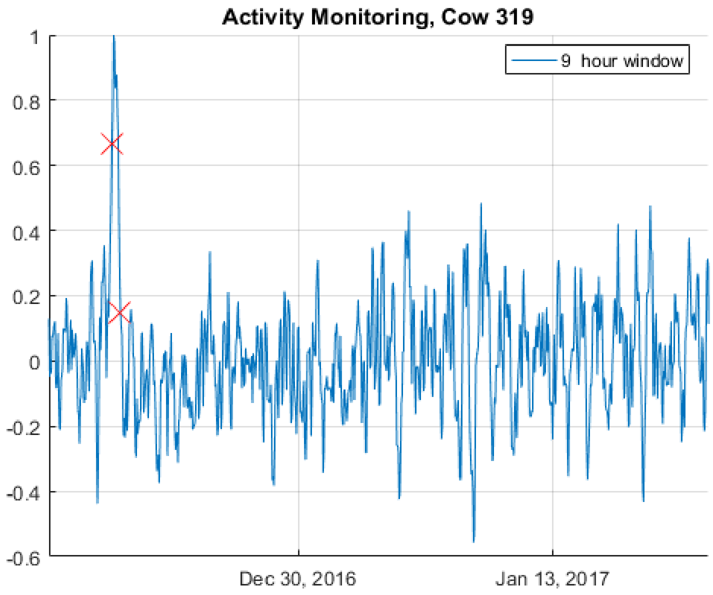

Figure C1: The plot of the results of the activity measurement algorithm of cow 319. A true positive estrus is detected and no false positive or false negative detection occurred. However, there is a lot of variation when not in estrus. Thus, the risk of false positive and false negative results is plausible.

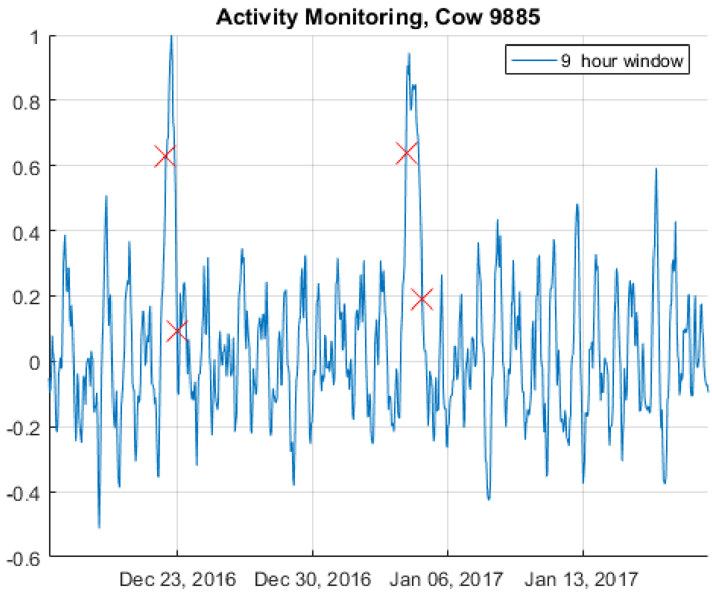

Figure C2: The results of activity measurement of the cow 9885. Both of the estruses are detected. However, the difference between the estruses and the rest of the period is insignificant as it was with cow 319. Thus, the possibility of false positive and false negative results exists.

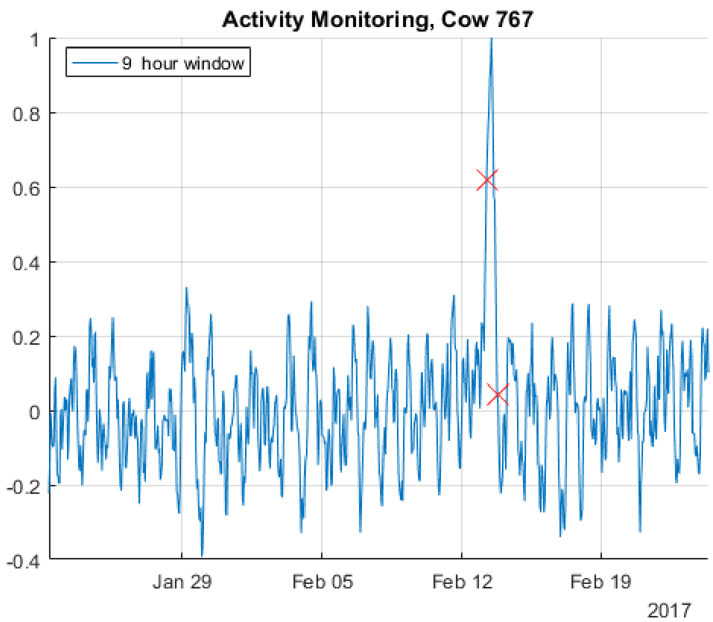

Figure C3: The plot of the activity measurement results of the cow 767. A true positive estrus is detected and no occurrence of false positives of false negatives. Additionally, the difference between proestrus and rest of the period is obvious. The risk of false positive or false negative results is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

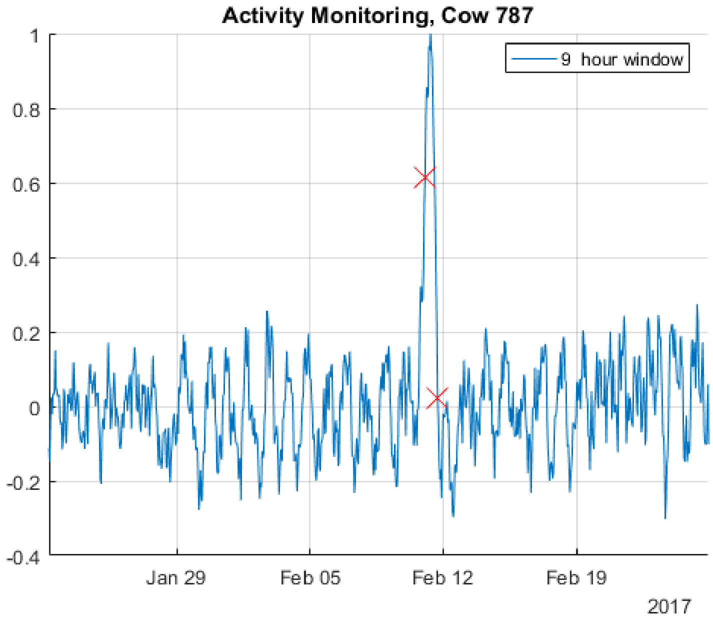

Figure C4: The results of activity measurement of the cow 787. The difference between the estrus and rest of the period is most distinct within this algorithm. Thus, the risk of false positive and false negative results is least significant.

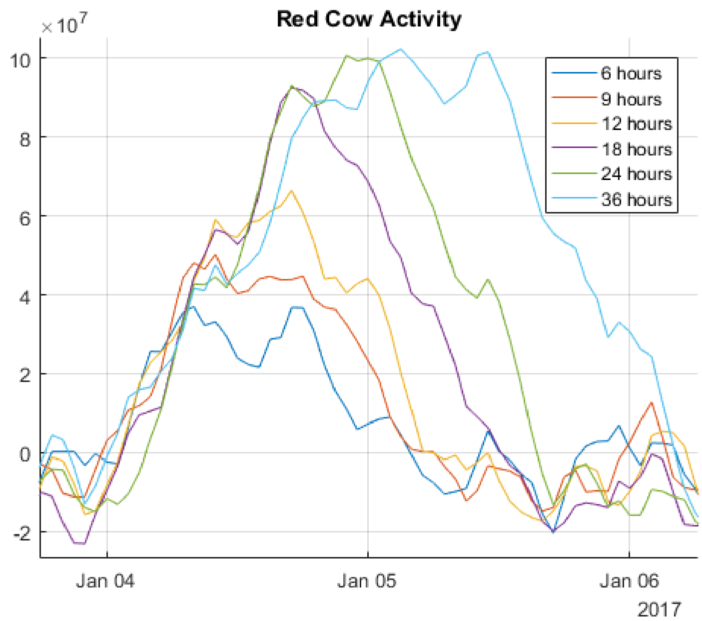

Figure C5: The activity plots of the second proestrus of the cow 9885. The figure illustrates the affect of varying the size of the integration window. Wider window increases the amplitude of the estrus and eases the detectability. However, simultaneously it delays the moment of detection.

D Variance Detection Full Results

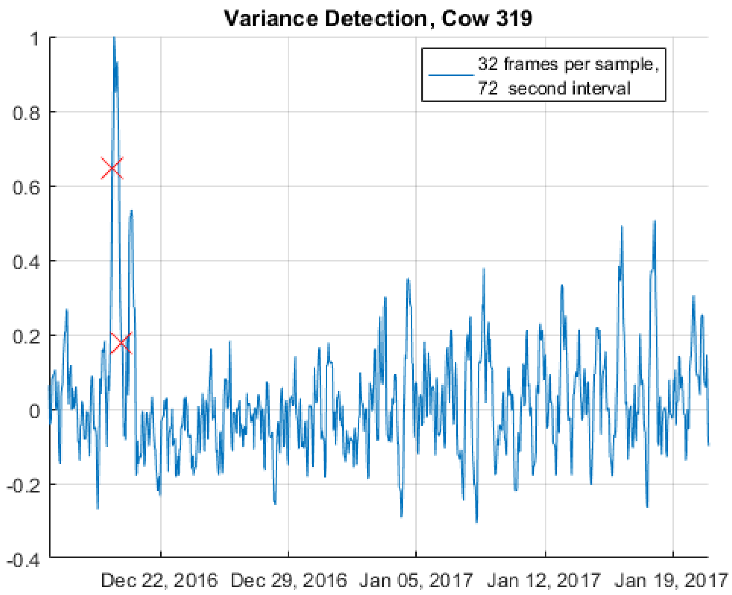

Figure D1: The results of the variance detection algorithm of the cow 319. The estrus on December the 20th is barely detectable. Additionally, algorithm yielded several false positive estruses. Furthermore, the amplitude of the false positives exceeds the true positive.

Varying the sampling interval affect the results:

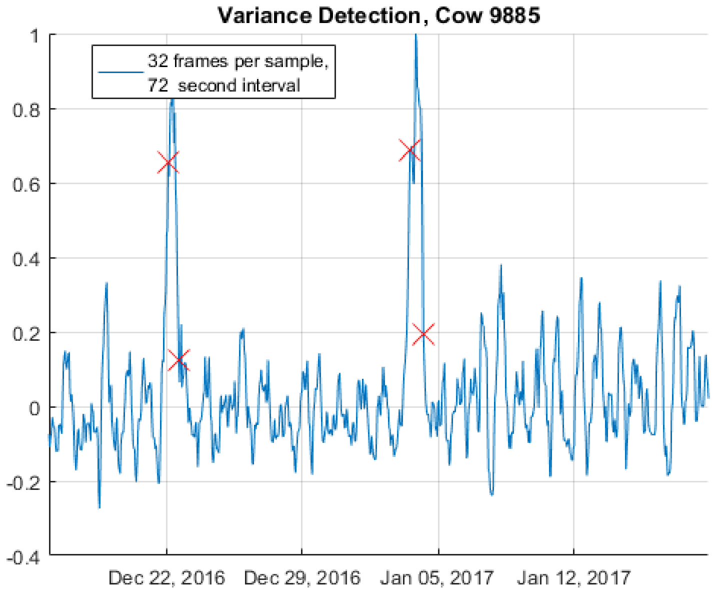

Figure D2: The results of variance detection algorithm of the cow 9885. The true positive estrus is detected on December the 22st. Furthermore, the amplitude difference between the detected estrus and other period is significant. However, the algorithm yield a false negative on January the 4th.

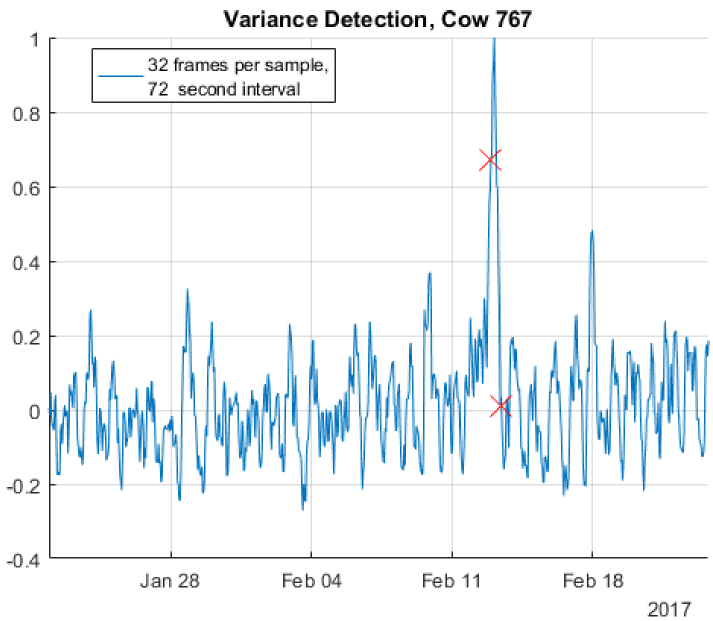

Figure D3: The variance detection results of the cow 767. There are multiple false positive detections in the data period. In general, there is no obvious difference between the estrus and non-estrus periods. Nevertheless, a true positive estrus is detected on February the 13th.

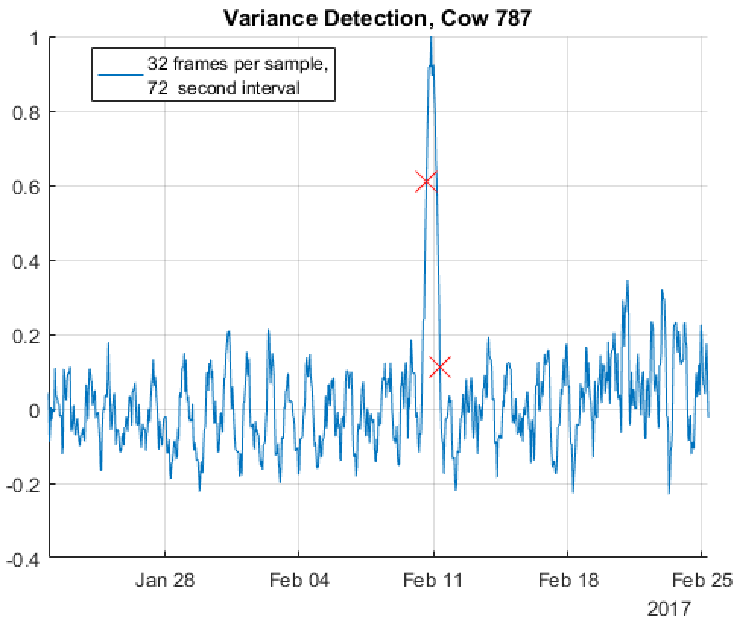

Figure D4: The results of variance detection algorithm of the cow 787. The true positive estrus is detected on February the 10th. However, a false positive detection occurred on February the 21st. Otherwise, the amplitude difference between estrus and non-estrus periods is obvious.

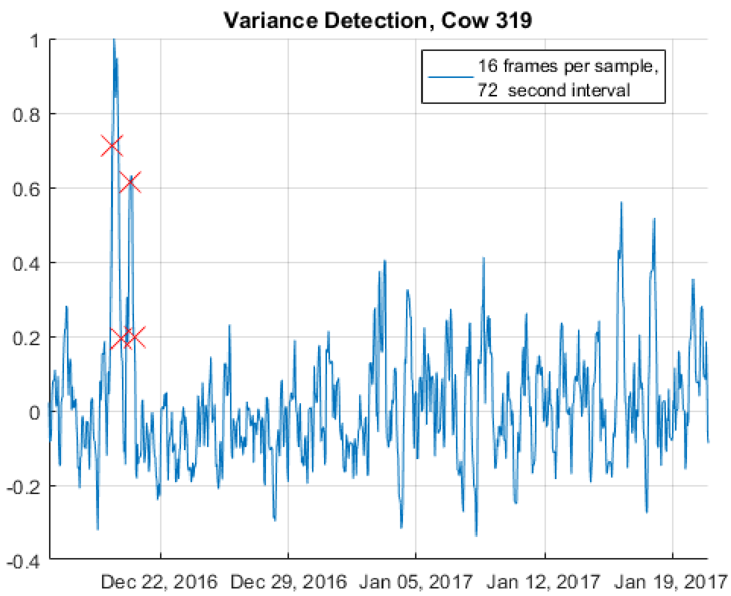

Figure D5: Reducing the sample size to 16 frames per sample and keeping the sampling interval in 72 seconds yields false positive results with the cow 319

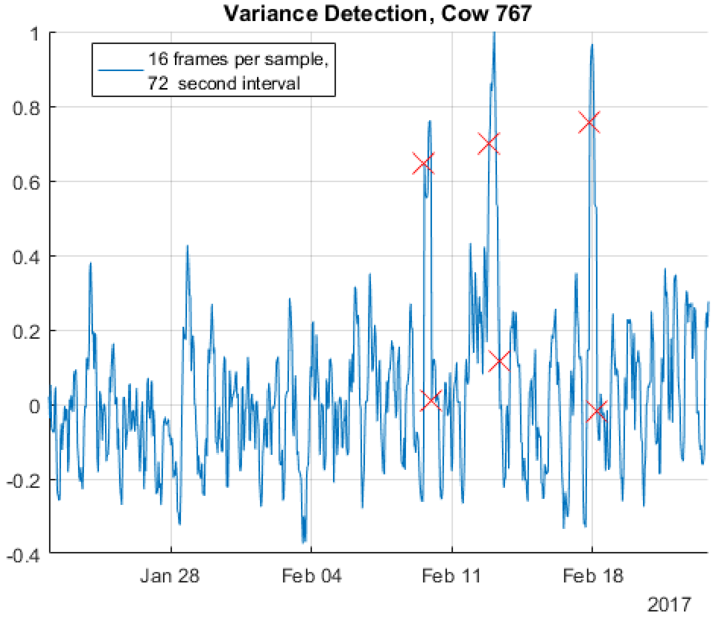

Figure D6: Reducing the sample size to 16 frames per sample and keeping the sampling interval in 72 seconds yields false positive results with the cow 767

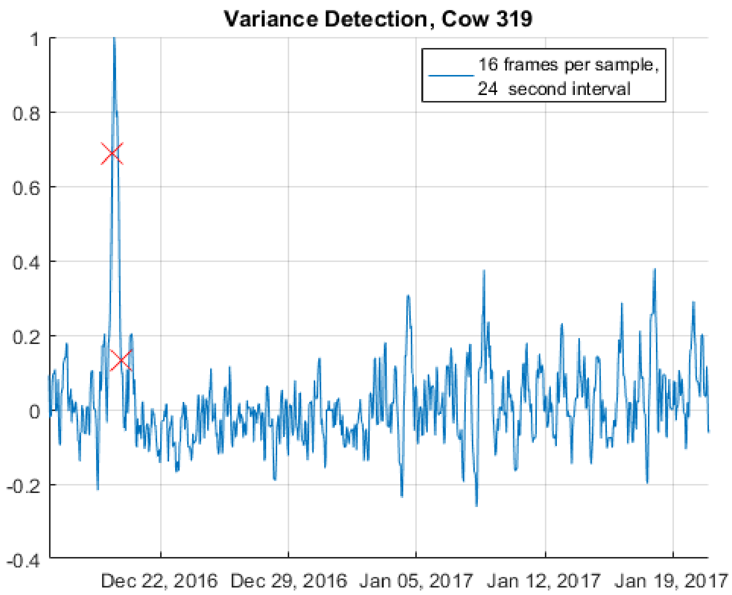

Figure D7: Reducing the sample size to 16 frames and increasing the sampling interval to 24 seconds improves the performance of the variance detection algorithm. Consequently, no false positive results occurs.

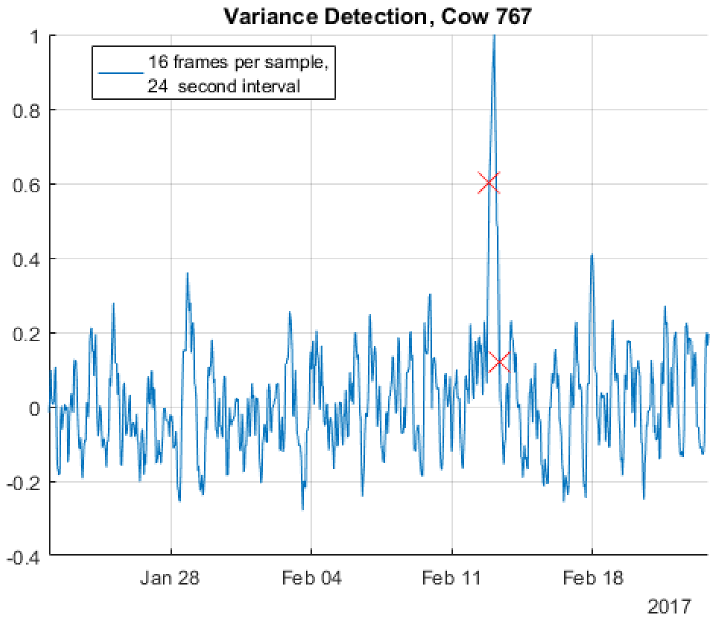

Figure D8: Reducing the sample size to 16 frames and increasing the sampling interval to 24 seconds improves the performance of the variance detection algorithm. Consequently, no false positive results occurs.

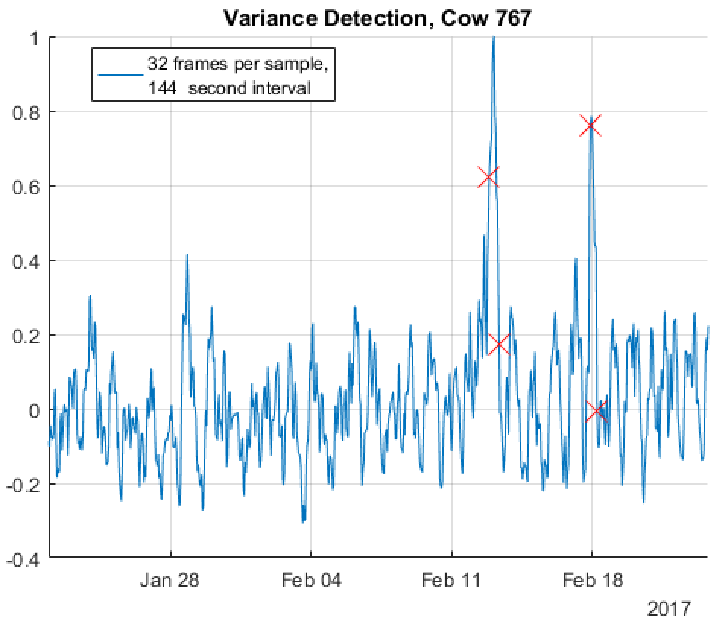

Figure D9: More seldom sampling yield false positive results as seen in this picture.

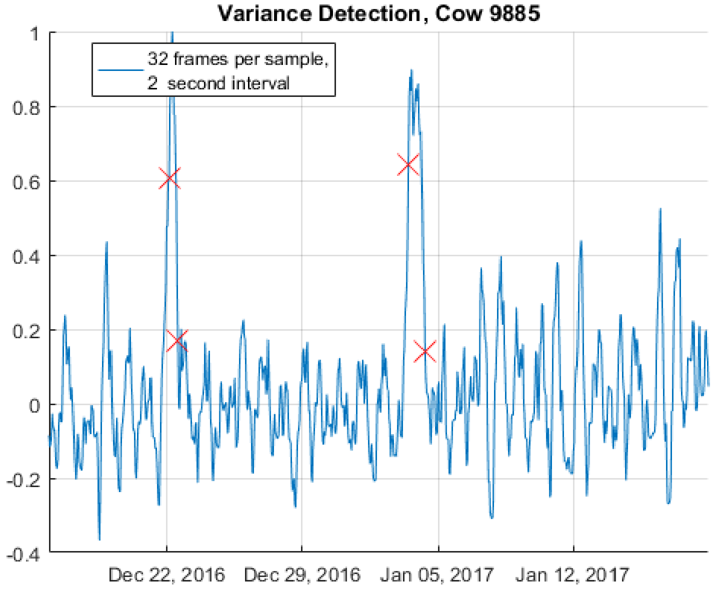

Figure D10: Decreasing the sampling interval does not directly improve the results as it is with the cow 9885. This sampling frequency corresponds approximately continuous sampling. Nevertheless, the results are worse than formerly.

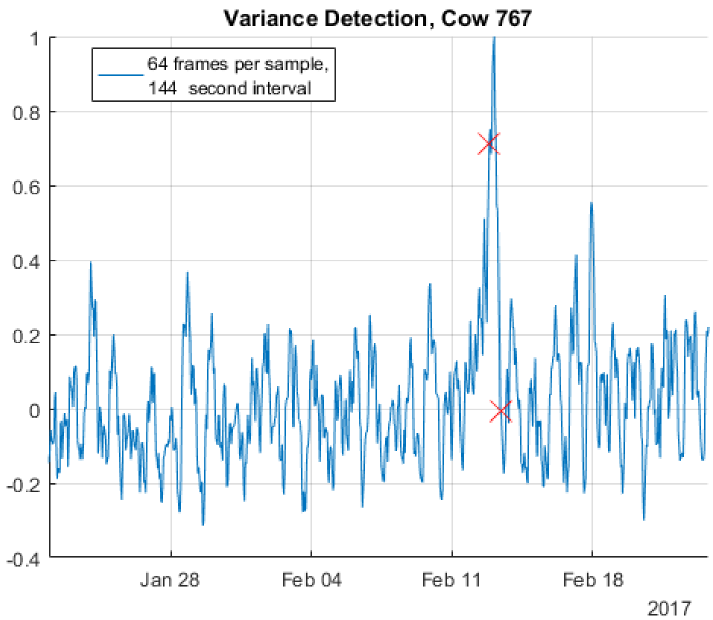

Figure D11:

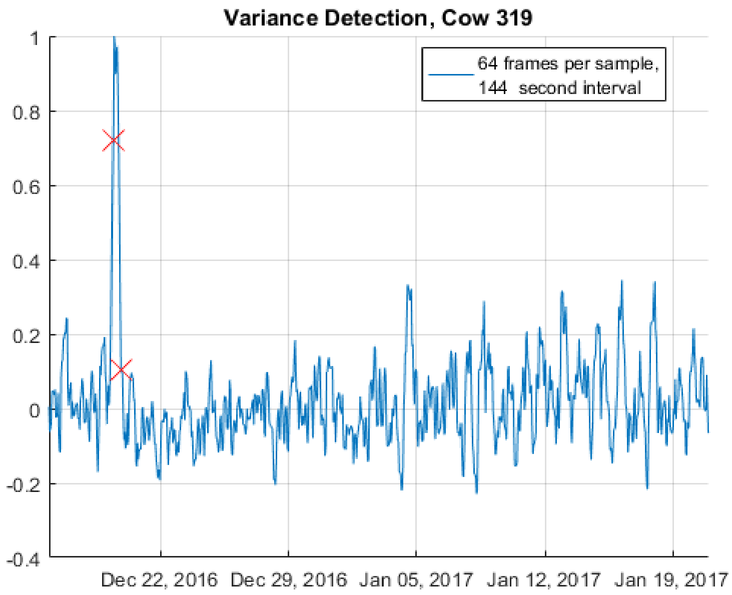

Figure D12:

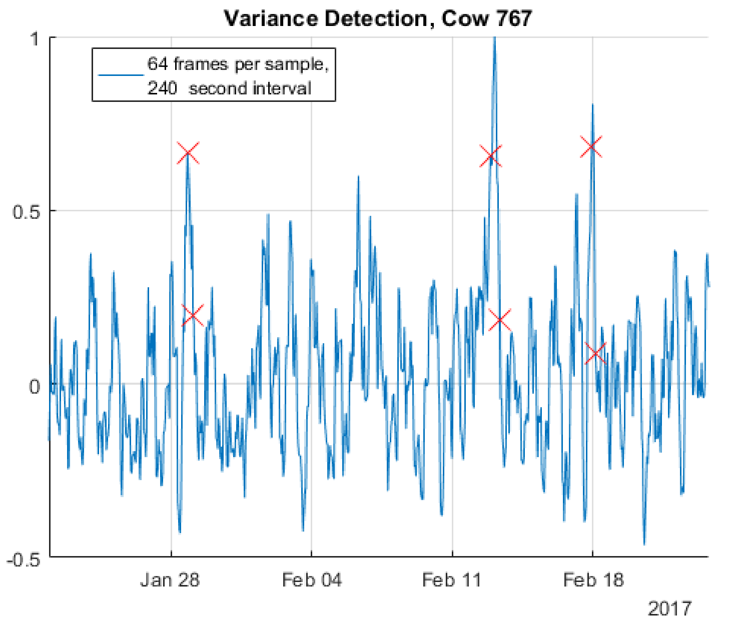

Figure D13:

E Inactivity Detection Full Results

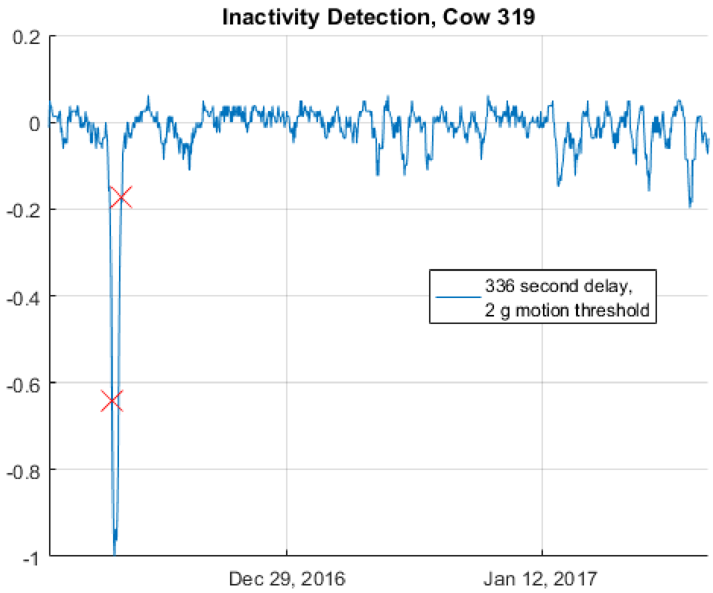

Figure E1: The results of the inactivity detection algorithm of the cow 319. The parameters are 336 second delay and $2\,\mathrm{g}$ motion threshold. A true positive estrus is detected on December the 20$^{\text{th}}$. The amplitude of proestrus and non-estrus is significant.

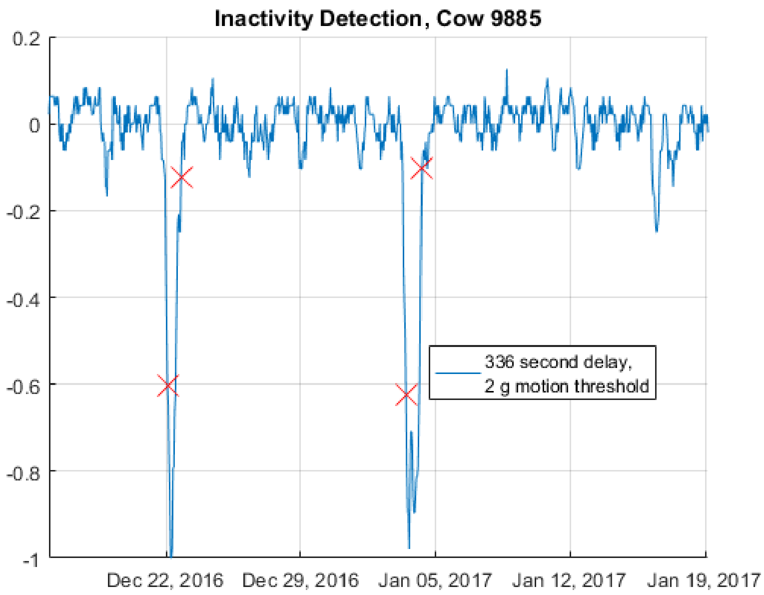

Figure E2: The results of inactivity detection algorithm of the cow 9885. The parameters are 336 second delay and 2 g motion threshold. Both of the estruses are detected as true positive on December the 22nd and January the 4th. No false positive or false negative detection occurred. The amplitude difference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

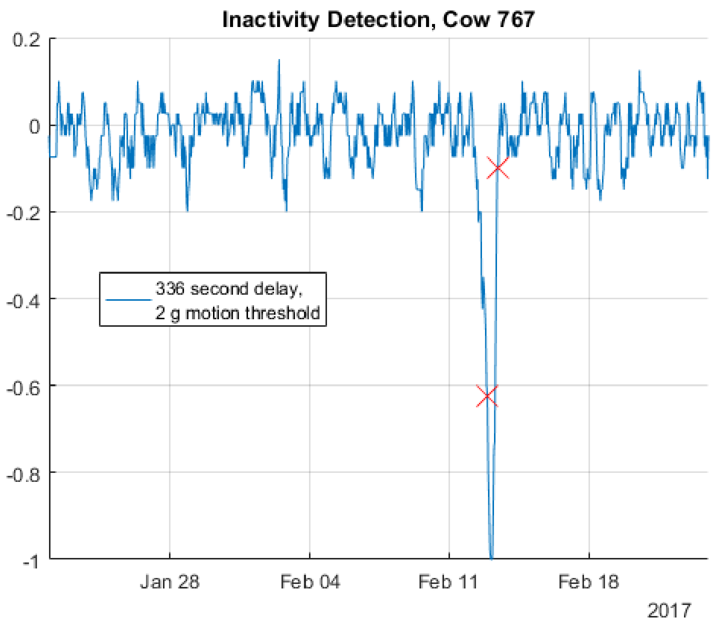

Figure E3: The results of the inactivity detection algorithm of the cow 767. The parameters are 336 second delay and 2 g motion threshold. A true positive estrus is detected on February the 13th. Additionally, the amplitude of the proestrus differs from none-estrus significantly.

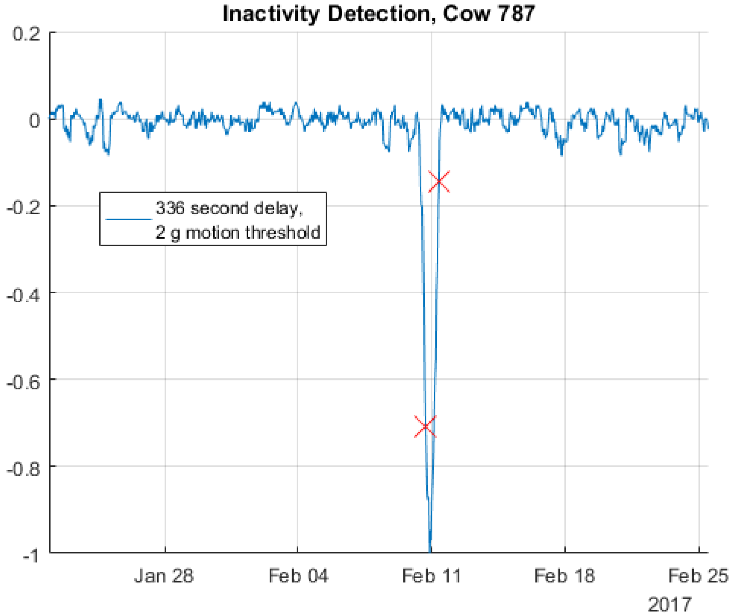

Figure E4: The results of inactivity detection algorithm of the cow 787. The parameters are 336 second delay and 2 g motion threshold. A ture positive estrus is detected on February the 11th and no false positives of false negatives occurred. Furthermore, the difference between estrus and other periods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

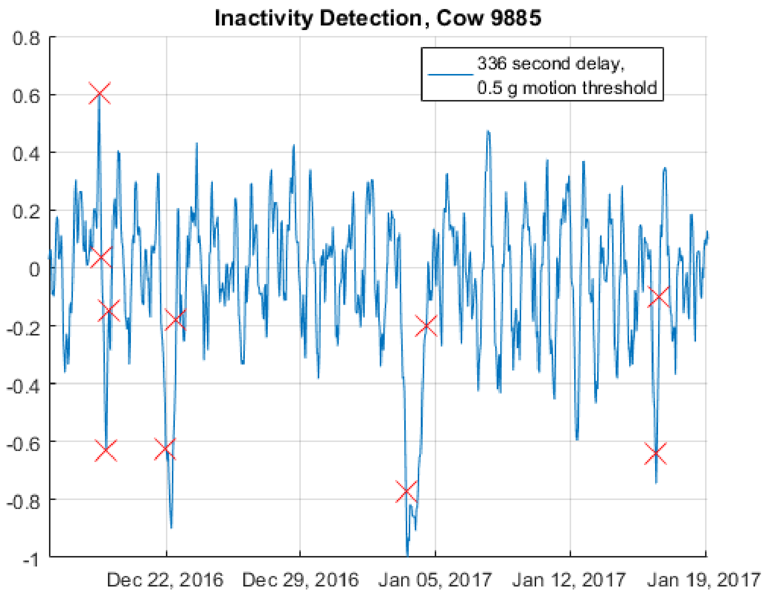

Figure E5:

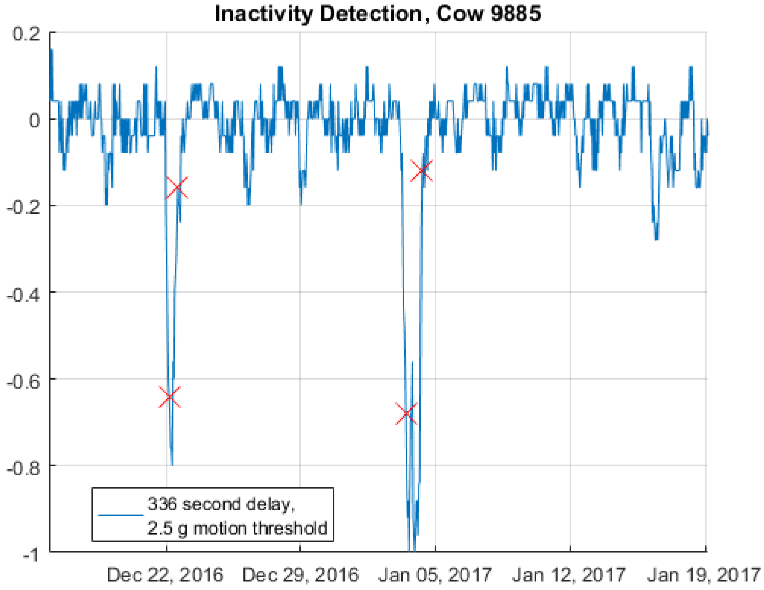

Figure E6:

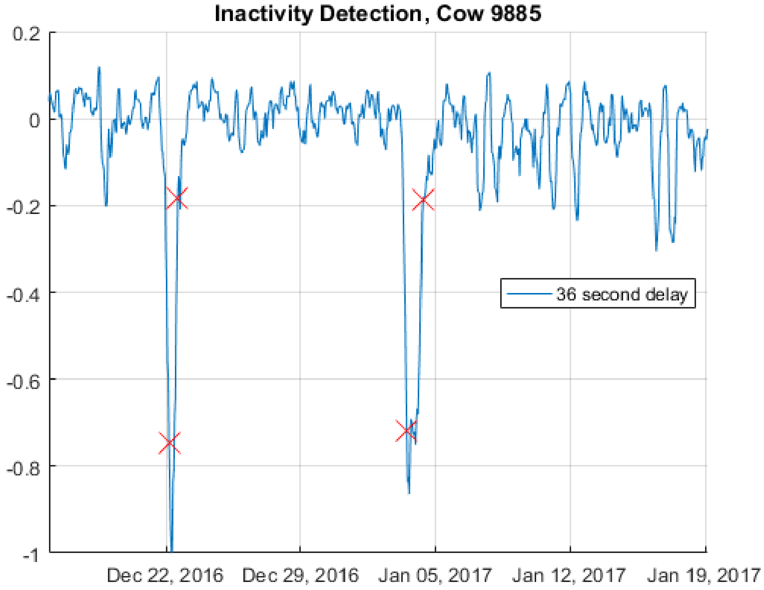

Figure E7:

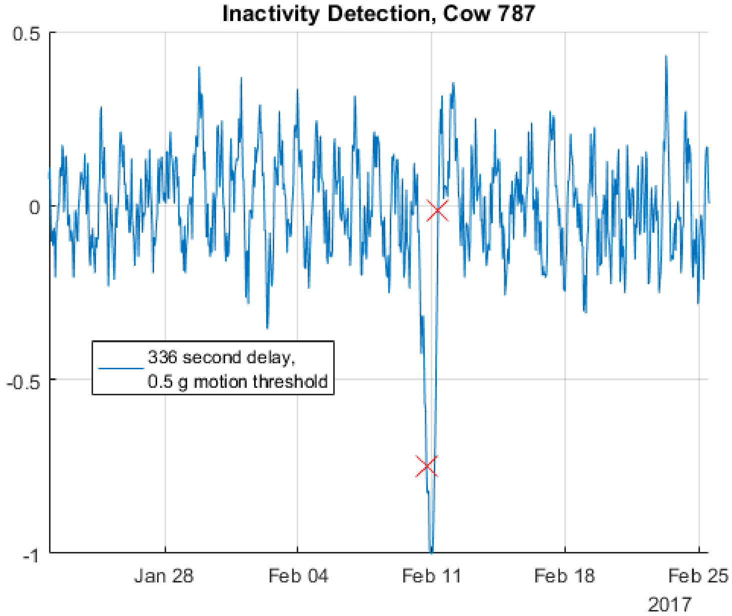

Figure E8:

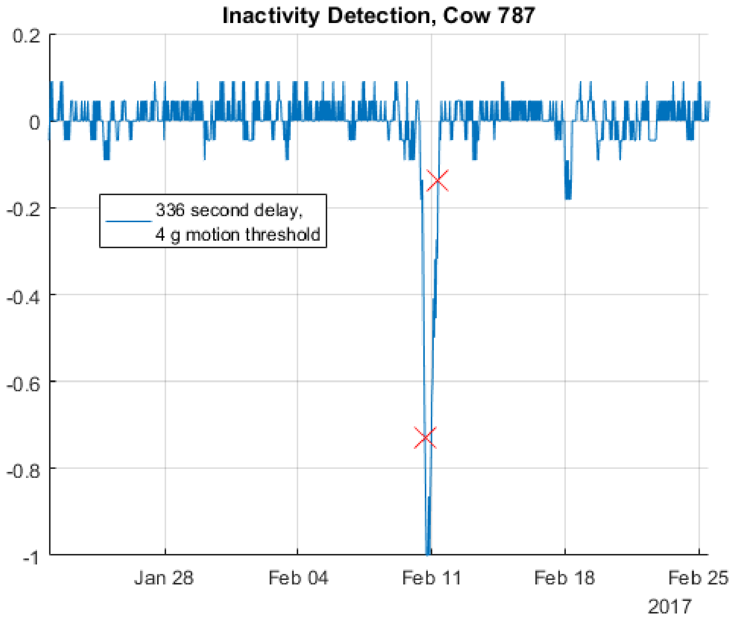

Figure E9:

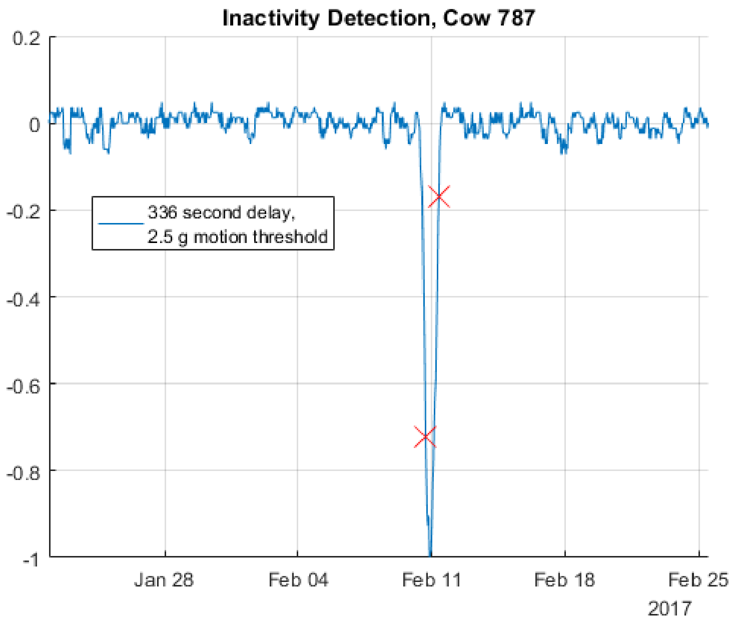

Figure E10:

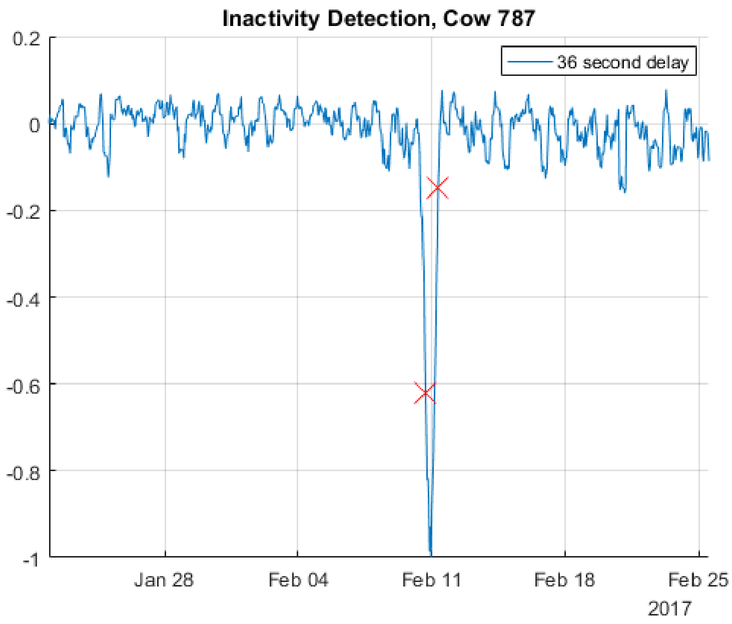

Figure E11:

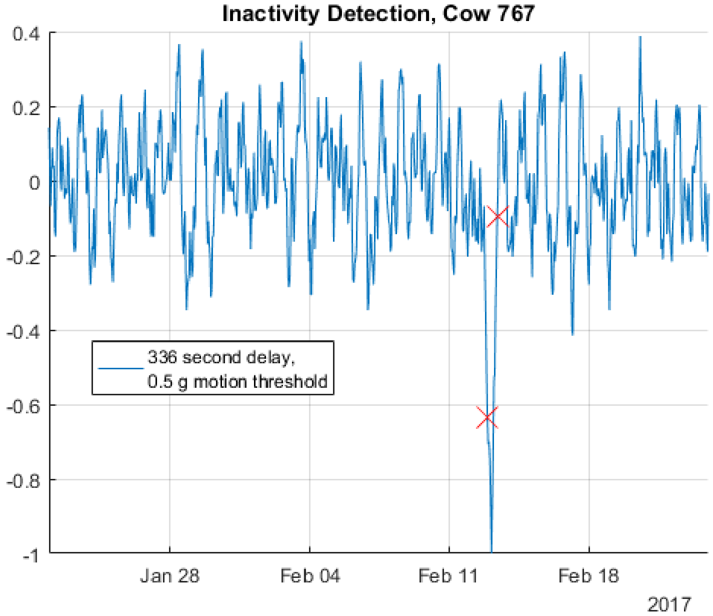

Figure E12:

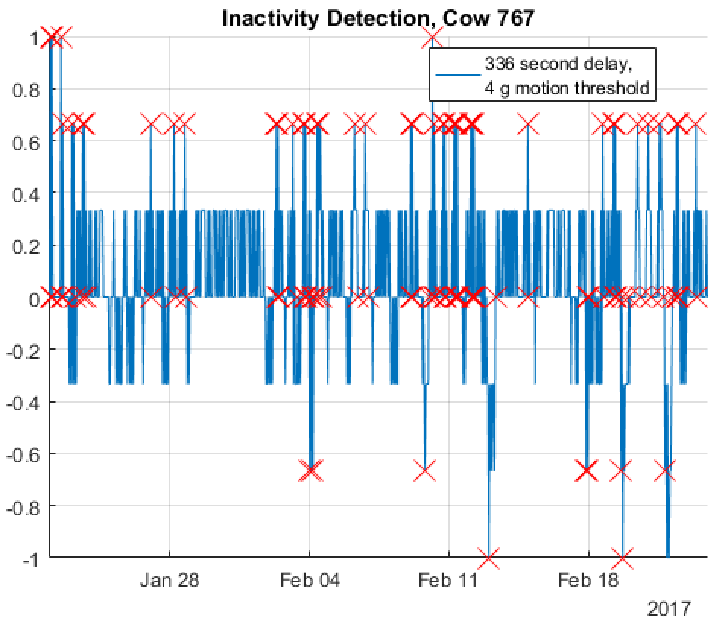

Figure E13:

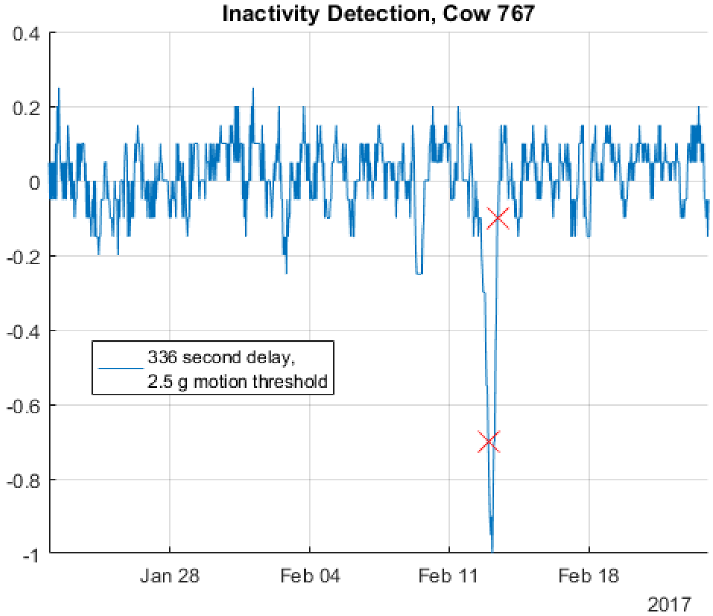

Figure E14:

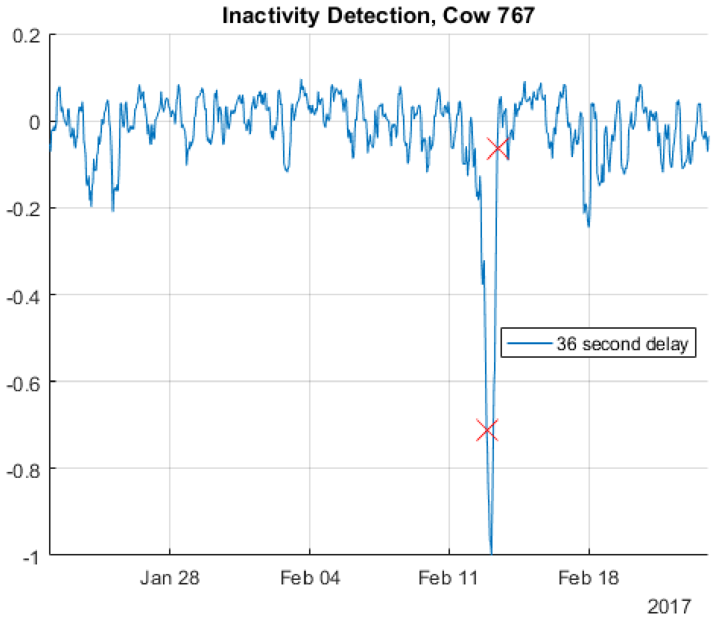

Figure E15:

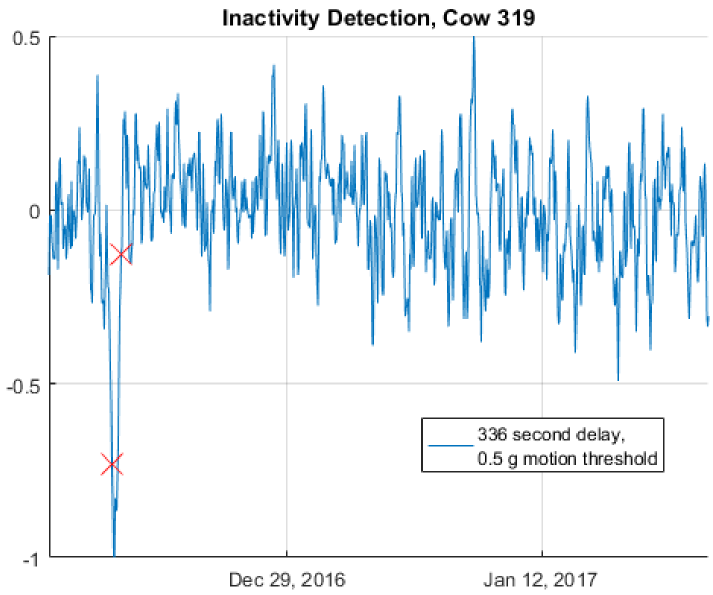

Figure E16:

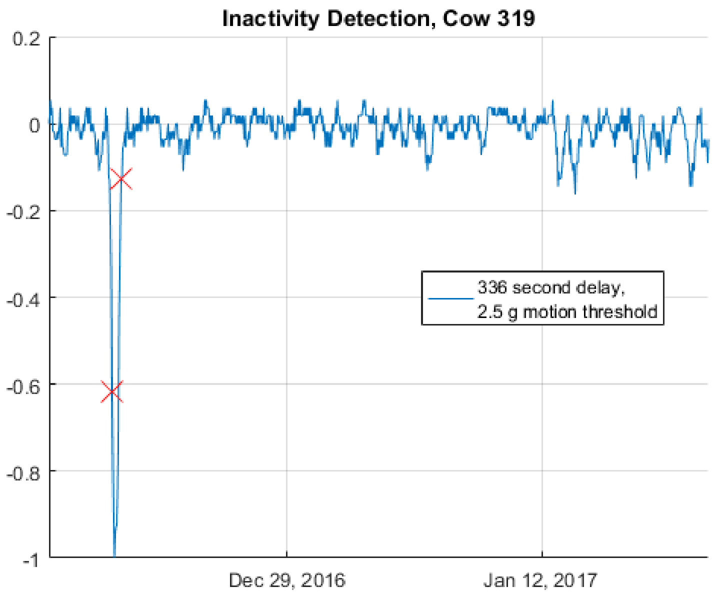

Figure E17:

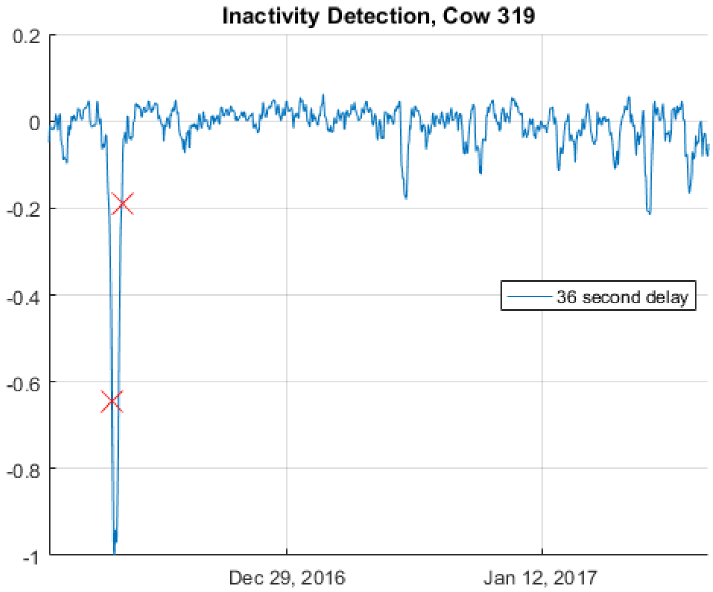
Inactivity Detection, Cow 319
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
36 second delay
Dec 29, 2016
Jan 12, 2017

Figure E18: