Automatic Weighing for Dry Stock

A thesis

submitted in fulfilment

of the requirements for the degree

of

Master of Science

at

The University of Waikato

by

GRZEGORZ ŚMIAŁOWSKI

KO TE TANGATA

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2013

Abstract

The aim of the research was to develop an accurate weight estimation algorithm for dry stock cattle using unsupervised walk-over equipment and to investigate the accuracy of the dairy cow algorithm for dry stock animals. Several customers and research organisations requested a suitable product for weighing dry stock, similar to the automatic weighing system for dairy cows. The development of the algorithm had to take into account the erratic behaviour of dry stock cattle. An additional requirement for the research was to improve cost effectiveness of the existing system, by removing the need for the lead-on platform.

Data was collected from a series of dry stock herds, as they walked over the existing dairy cow walk-over weighing (WOW) platform, in either a paddock or a stockyard. All dry stock cattle were static weighed before walking over the WOW to obtain an accurate true weight of the animal. The weight estimations produced by the dairy cow algorithm were recorded and the raw load cell data was captured. The raw load cell data was then used to devise a new algorithm specifically for dry stock cattle. The new algorithm was tested against the collected load cell data from various herds. The dairy cow algorithm results were used to compare the accuracy of the existing algorithm on dry stock cattle.

The new devised algorithm used a threshold to locate a rough start and end point for each walk-over event. The data between the initial start and end points was processed to locate the period which had the full weight of the cattle located on the weighing platform. The algorithm estimated the weight of the animal from the data between these points.

It was found that the dairy cow algorithm was nearing the required accuracy, but was not able to determine weights for the majority of the animals. The new algorithm devised for dry stock cattle was unable to obtain an improvement in accuracy, but was able to estimate weights for a greater proportion of the cattle. Additionally the new algorithm was successful in eliminating the need for the lead-on platform for detecting animals.

Acknowledgements

I would like to thank my supervisor, Mark Apperley, for his guidance throughout the project, and his thoughts and suggestions. Having the input of another person made getting through difficult situations easier.

I would like to thank Gallagher's for giving me this opportunity to work on this project. I would also like to thank all the staff that I had the pleasure of working with throughout my time at Gallagher's.

I would like to thank Owen Boyes for taking it upon himself to be my mentor at Gallagher's. His knowledge was invaluable. The knowledge he gave me helped guide me throughout the project and he was very helpful whenever I required resources for the project.

I wish to thank Tony Smith, for organising this project, and organising the funding.

I have to thank Graham Smith and Diana for the use of their farm and animals. Thank you for the large amount of time which you have been able to put towards the project, and for the continued support even when facing problems that were out of our control.

Funding

The funding for the project was provided by the Ministry of Science and Innovation, under the TechNZ initiative. The aim of the funding is to create innovative systems which will advance technology in New Zealand.

Hardware

I would like to thank Gallagher Global Limited for supplying the hardware used during the project. The support from the technical staff was greatly appreciated also.

These items included; walk-over weighing crate, laptop, PC in a plastic case, DairyScale equipment, SmartReader, video camera, gates and gate hardware.

Table of Contents

List of Figures

List of Tables

1.0 Introduction

Computerisation has become common in all fields of industry, allowing everyday tasks to be completed in a faster, more efficient and automated way. The same trend exists in agriculture industry, with the use of the automatic weighing system for dairy cows being just one example. The weighing system for dairy cows system works well in the constrained environment of a dairy shed, so there was a demand for a similar system to be used in paddocks for dry stock.

The need for automated weighing of dry stock cattle was requested by research organisations and clients from several countries. They were aware of an existing weighing product for dairy cows that estimates the weight of an animal walking over the platform without stopping. The differences in behavioural patterns between dairy cows and dry stock cattle (bulls) lead us to believe that the algorithm used for the existing product will need to be modified for the purpose of weighing dry stock cattle.

Gallagher Global Limited (Gallagher's) is a company that manufactures a range of products; including animal management systems, fuel systems, and security products. Gallagher's have been making agricultural equipment for over 70 years, and at first become noticed for their electric fence development. Gallagher's have a range of weighing products and EID (electronic animal identification) systems. The weighing systems range from static weighing products to the DairyScale. The DairyScale is similar to a static weighing system, except the cow can walk over the platform without stopping and the weight is calculated. The DairyScale algorithm was designed by Paul Teal (Industrial Research Limited) and was implemented on the existing static weighing hardware, with slight modifications.

1.1 Brief Look at the Project

The aim of this research was to identify a method for obtaining accurate weight estimations for dry stock cattle using a walk-over platform. It appeared that the existing DairyScale hardware could be used without alterations, but the algorithm may require modification or a new algorithm need to be designed to take into account the behavioural differences between dairy cows and dry stock cattle.

The approach taken was to use the existing system to record raw weight data (data from the load cells without any modification) and also use the existing algorithm to estimate weights. This was done to identify how accurately the existing algorithm for dairy cows performs when used to estimate weights for dry stock cattle. The new algorithm specifically designed for dry stock cattle was deduced from the raw data collected from the dry stock, and the results compared to the dairy cow algorithm.

1.2 Definition of the Problem

Several customers and research organisations have asked if they can use the current DairyScale product (designed to weigh dairy cows) for weighing dry stock cattle.

Gallagher's currently has a WOW (Walkover weighing) product which is designed to weigh dairy cows. The problem is that dry stock cattle behave differently to dairy cows. The dairy cows are placid animals that walk fairly calmly most of the time when leaving the milking shed. They are used to going down a race where the weighing platform is installed as part of the race.

Dry stock cattle are usually grazing for long periods of time and are not used to races, tight space, or other metal objects. These situations make them behave erratically which make them difficult to weigh.

These differences in behaviour have to be taken into account when designing a new algorithm, or when altering the existing algorithm.

1.3 Motivations

There has been significant interest from end users and the research community for a system which can automatically weigh dry stock. They have seen the DairyScale product (WOW for cows) and have been asking if they can use it for their application, of dry stock animals. The DairyScale system allows dairy cows to get weighed while walking over a platform, without the supervision of a person. It is designed to be installed in the exit race of a rotary dairy shed.

Gallagher's cannot guarantee that the results of the DairyScale will be reliable for weighing dry stock cattle, as the behaviour of dry stock animals is significantly different to dairy animals. The project aims to test the DairyScale algorithm (which is designed for cows) on beef cattle and record raw data.

There are several groups that would benefit from a product like this. Large scale dry stock cattle farmers would find this system advantageous for monitoring farm outputs closely. This method of weighing would be less time consuming and require less labour by the farmer, compared to other systems. Agricultural research organisation would find the WOW system very useful for frequent weight checks of dry stock cattle. It would be a quicker to weight them and be less stressful for the cattle.

It is important for Gallagher's to have a dry stock WOW system to gain market advantage as a weighing innovator. Competitors' WOW products do not use a lead-on platform, and that makes the DairyScale less competitive. Mark Harris (Gallagher's) has suggested that he would like to have a WOW product which does not require the lead-on platform, so the price of the WOW product can be decreased. Although the main aim of this project is to create an algorithm for weighing cattle, it would be an advantage to eliminate the need for the lead-on platform.

1.4 Other Systems

The most common and accurate weighing system for dry stock is a static weighing system. That requires the animal to be stopped, and held in a crate which is mounted on load cells. The scale head takes a short period of time to get a fix on a weight, and then the farmer can release the animal by opening a gate.

Gallagher's has a walk-over weighing system currently, which was designed to weigh dairy cows. The hardware was suitable for this project; however the algorithm may not be suitable to be applied to estimate dry stock cattle weights.

1.5 Outline

The problem posed for this research was a need for a system to weigh dry stock cattle without supervision. Gallagher's currently has a product for weighing dairy cows, so clients wanted to know if the existing system can be used to weigh dry stock cattle. It was suggested that the differences in behaviour of dry stock cattle and dairy cows could pose some problems with using the existing algorithm to estimate weights for cattle.

The first stage of the research was dedicated to becoming familiar with the existing systems, before gathering data in the field. In the second stage a series of trials were conducted with the dry stock cattle. These data collection sessions recorded raw load cell data and produced DairyScale weight estimations. Based on the raw load cell data collected a new algorithm was devised for dry stock cattle. The new algorithm was used in the lab to estimate weights from the data collected previously from animals walking over the WOW, and the results were compared with the true weights of the animals. In the third stage the specifically new algorithm would be implemented on the DairyScale hardware and tested, to assess its performance and accuracy.

The output of the research is primarily to identify if the DairyScale algorithm can weigh dry stock accurately. Secondly, to produce an algorithm specifically designed to weigh dry stock cattle. Then test each algorithm to see if it can achieve weight estimates within 3%, for 80% of the recorded estimated. A possible bonus output of the project would be to remove the need for the lead-on platform.

1.6 Limitations

There are several limitations that need to be taken into account for this project, from the hardware limitations to the amount of data that can be collected during the project time limit.

The complexity of the algorithm must be kept to a minimum, as the processing power of the DairyScale unit is limited. Therefore the hardware would not be able to run complex AI (artificial intelligence) methods.

The length of time to complete the project is limited to 12 months. Several tasks need to be achieved during this time; data collection, creating an algorithm, testing the algorithm and writing up a thesis. To make sure the project is completed within time a timeline must be followed (see Appendix B – Project Proposal).

We want to emulate the way the farmers would usually use the WOW (in the paddock), when cattle is only weighted periodically. The farm available for the research had only a limited number of animals available. To overcome shortcoming we had run the same herd over the WOW two or more times to increase the number of events captured. The problem with this is that if we use the same herd too many times, they do start to remember the equipment and behave differently.

The weather could have an impact on frequency of data collection sessions. The biggest obstacle would be long wet periods causing the fields to be easily damaged by moving cattle. The cattle could not be moved between the fields for weighing during or after these weather conditions.

The equipment that was used is not something that most people have experience with. The knowledge of the equipment will have to be learnt by reading manuals and talking to Gallagher's technical staff.

1.7 Overview

Chapter 1 provides an introduction into the research topic. Providing the reasons why and how the project is carried out.

Chapter two discusses the background of the project. Provides details on how the existing WOW system functions for weighing dairy cows. The chapter includes information on other methods of weight estimation for cattle and information on similar products. It discusses the constraints of the supplied hardware and the challenges posed by dry stock compared to dairy cows.

Chapter three contains information about the methods and the equipment that were used to gather data. Provides a detailed description of how the data was analysed to devise an algorithm for dry stock and describes how the algorithm works.

Chapter four focuses on the implementation of the sum threshold method for estimating weights. It discusses the results produced by the new algorithm and compares them to the DairyScale results. It also includes the outcomes of the two additions to the new algorithm that were tested.

Chapter five discusses the findings of the project and suggests future research options. It examines the limitations encountered during the research and provides a summary of the project.

2.0 Background

This chapter reviews and discusses the background and literature relevant to this research, including time series analysis.

It describes how the Gallagher's dairy cow WOW (walk-over weighing) product functions and its components, and discusses the system's constraints. Other products similar to the DairyScale are described. Discusses other methods of weighing and lists the various reasons for weighing cattle.

2.1 Review of Relevant Literature

2.1.1 Time Series Analysis

This section explains a mathematical method used throughout the research to process time series data. This data was processed using time series analysis to extract meaningful characteristics and information from data collected, from the dry stock cattle crossings of the WOW.

Moving window analysis is one of the techniques suitable for analysing time series data. The main feature of this technique is its ability to smooth out the highs and lows, allowing us to see the trend in the data. A window of fixed length is chosen, the length can be of any length, but a minimum of two. An average is calculated for the data points within the window, and recorded. The window moves forward one position in the time series. In the next position the average is re-calculated and recorded. This process is repeated for the rest of the data points, or of the section of interest.

This method is useful for data involving animals moving because it smoothes the peaks and troughs caused by the body weight shifting as the animal takes steps on the platform. An example of the peaks and troughs that occur as animals walk over the weighing platform is shown in Figure 1within the rectangle.

Figure 1: Fluctuations of weight.

Figure 1 note: The rectangle shows the animal weight fluctuation as the animal walk, over the weighing platform. Both the front and rear load cells are combined in the thick blue trace. The two spikes in the bottom trace indicate times when the RFID tags were read. Y-axis is in kilograms. X-axis is time at a rate of 41.667Hz. Thin red trace: RFID tag read. Dashed red line: Threshold (280kg).

2.1.2 The DairyScale System

The current WOW (walk-over weighing) system is called DairyScale; it is produced by Gallagher's. The algorithm was designed to weigh dairy cows. The DairyScale WOW crate has two platforms, a main and a lead-on. The main platform has two load cells under it, a leading and trailing and the lead-on has one load cell (see Figure 2).

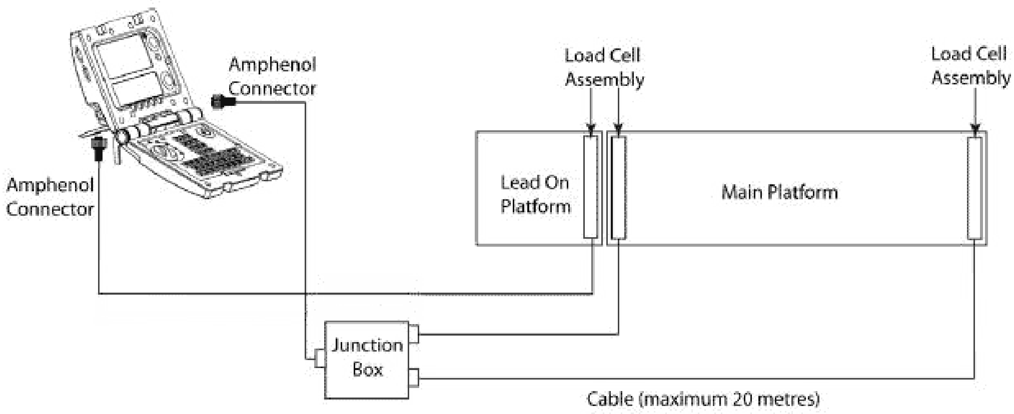

Figure 2: Scale head with load cell platform layout

(Gallagher Group Limited, 2011)

The current system detects edges from the load bars under the main platform to determine the location of the animal. The lead-on platform is used to detect if there is another animal following close behind, and will disregard both animals if they are too close to each other. The lead on platform is not used to determine the animals' weight.

The DairyScale automatically selects the best filter to estimate an animals' weight, depending on the animals walk across the platforms. There are 35 filters with varying lengths for the DairyScale to choose from, ranging from 400ms to 3 seconds (Teal).

The DairyScale system has an error checking system, which is only accurate after an animal has been seen repeatedly (12 times). It uses the animals' previous records to decide if the weight estimation is within a certain range to decide if the new estimate is accurate. This part of the error checking system will not be suitable for beef cattle, as the frequency of weighing is expected to be less frequent.

2.1.3 Other Animal Weighing Systems

There are other systems on the market for weighing dairy cows. In New Zealand the competitor to Gallagher's DairyScale product is the Tru-test WOW! XR3000 (Tru-Test Group). Another company with a similar product is Westfalia, with the Taxatron 5000 (GEA Farm Technologies). These systems are designed to weigh dairy cows so are a competitor to the DairyScale and not a system for dry stock cattle.

Tru-test WOW! XR3000

Tru-test produces a WOW product called the XR3000, specifically designed for dairy cows (Tru-Test Group) like the DairyScale.

The competitive advantage of the XR3000 is that it is sold at a lower price point, compared to the DairyScale. This is because the XR3000 does not have a lead-on platform, whereas the DairyScale has this added component causing the system to cost more.

Taxatron 5000

The Taxatron 5000 is a WOW system for dairy cows, from Westfalia (GEA Group). The WOW system is available as part of an integrated dairy shed system (GEA Farm Technologies).

The DairyPlan C21 (propitiatory software) allows the weight data to be integrated with other dairy systems on the farm. The software is able to give the farmer reports on the relationship between feeding and milk yield. That allows the farmer to make informed decisions on feeding strategies.

2.2 Weighing Background

Livestock owners need to weigh their animals from time to time for a variety of reasons. Cattle farmers have several methods they can use to estimate the weight of their animals. Some of these methods are easier than others. Weight can be estimated by measuring the animals dimensions, using a cattle calculator, or the most common way nowadays is to static weigh them. The problem with these methods is separating each animal and the time consumed to weigh each animal. A WOW system will help save time when it comes to weighing dry stock, as it has done with dairy cows.

Farmers may choose to weigh their animals for one of many reasons, be it to determine best time of sale, monitoring the health of the animal, dosing of medication, or farm management. Weighing is an important part of the farming process.

Some farmers use visual observation for these tasks, but it is usually very inaccurate and not recommended for use when determining rations, medication dosages, or for any purpose when an accurate weight is important.

Several methods farmers use to estimate weights of cattle are discussed in this section. Static weighing is by far the most accurate method of obtaining an animal's weight. It is also similar to the walk-over weighing method, as they both use load-cells. Three other methods of estimating an animal's weight are also described.

2.2.1 Static Weighing

A static weight is collected using load cells under a weigh crate, connected to a scale controller. Usually the animal walks down a race into a weighing crate, the famer shuts a gate behind them. The weigh is deduced by the scale controller. Sometimes there will be an ear tag reader, so the animals EID number can be stored against the weight. The animal is then released by the farmer after the weight is recorded.

Advantage:

- The farmer doesn't need to get too close to the animals during weighing (safety).
- Quicker than using a weigh tape.
- Very accurate.

- Digitally record weight and EID tag number.

Disadvantage:

- Stresses animal (being in a crate).
- Static weighing is labour intensive, compared to WOW (Alawneh, 2011).
- Greater cost of equipment.

2.2.2 Measuring Dimensions

A simple method of estimating an animal's weight can be done with a measuring tape and estimation formula. The heart girth and body length measurements are required for this weight estimation technique.

The method of performing this estimation is follows. See Figure 3 for a diagram of the points referred to in the method.

Method (Pater, 2007):

1. Measure the length of the body, from the point-of-shoulder (A) to the point-of-rump or pin bone (B).
2. Measure the circumference or heart girth (C). Measure from a point slightly behind the shoulder blade, down the fore-ribs and under their body behind the elbow all the way around. After these measurements are made in inches – use the following formula.
3. (Heart girth X heart girth X body length) divided by 300 = weight in pounds.

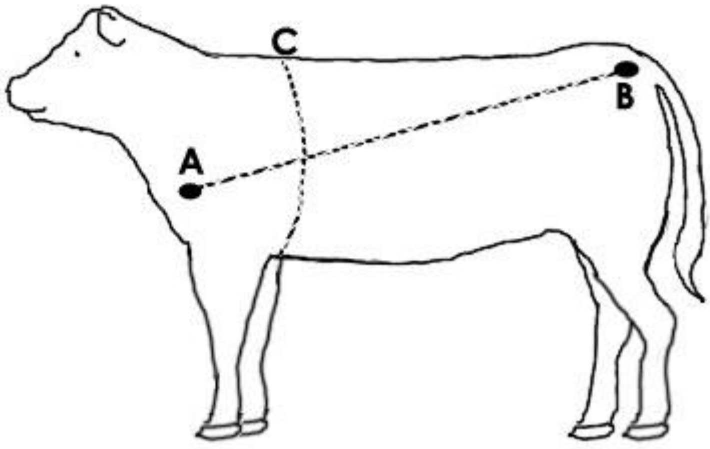

Figure 3: Cattle measurement. A: Point of shoulder, B: point of rump, C: Heart girth.

2.2.3 Chesterman's Cattle Gauge

The Chesterman's cattle gauge is used to estimate the weight of the animal's forequarters, not the live weight of the animal.

Chesterman's cattle gauge has a measuring tape and a circular slide rule which allows the user to find the estimated weight of the animal. The user measures the heart girth and body length (like the pervious method), then uses the circular slide rule to estimate the dead weight of the four quarters.

The instructions for the Chesterman's Cattle gauge are on the inside of the lid. They are as follows:

"First, take the length from the foremost upper corner of the shoulder blade bone, in a straight line to the hinder-most point of the rump by the tail, and next the girth close behind the fore-legs. The measures carefully taken will, with the assistance of the circular sliding plate, tell the dead weight of the animal's forequarters either in the London stone of 8 lbs. the country stone of 14 lbs. or the score of 20 lbs." (Chesterman, 1842)

The formula used by the calculator is: *Weight = 3.35 x Length x Girth squared*

Patent: James Chesterman of Sheffield in 1842.

2.2.4 Cattle Weigh Tape

Cattle weigh tapes use the circumference of the chest to predict the live weight of the animal. This simplifies the method covered earlier of measuring dimensions to estimate the animals' weight. Some weigh tapes have two different scales on the tape, one for stock of normal build and the other for fat stock (fattened for the production of meat) (Hyperdrug Pharmaceuticals Ltd).

Only one measurement needs to be taken, and the weight is marked on the tape. The cost of the tape is fairly cheap. The instructions vary depending on the manufacturer of the tape. A measurement of the animals' circumference is taken. It is either taken immediately behind the elbows or the hearts girth, which is about 5cm behind the front legs (King, 2012).

The portability of the equipment for this makes it a convenient tool that can be easily taken into the field.

2.3 Background Constraints

This section describes and explains some of the background constraints of this project. It will discuss the differences between dairy and dry stock, and some of the challenges of dry stock cattle. It will also cover the constraints of the hardware being used for this research.

2.3.1 Challenges of Dry Stock

Dry stock cattle usually spend most of their time in a field, without much human contact and without equipment. This means that they can be afraid of people and metal objects. They are usually in a large paddock so they can roam freely with lots of personal space and don't get moved often. This means they are not used to tight spaces, tight walkways, like races, similar to the WOW crate.

When cattle are scared or stressed they can move quickly or be stubborn, so if they are scared of the WOW it makes it difficult to weigh them. This behaviour can be observed during static weighing, often dry stock cattle will move around the crate trying to find a way out, as they are not used to being in confined places. The WOW should not be as intimidating, as both ends of the crate are open.

2.3.2 Dairy Cows verses Dry Stock Cattle

When dairy cows use the DairyScale system it is usually installed in the exit race (narrow metal railings) of a rotary milking parlour. They are accustomed to walking through races and do so calmly, so they do not find the WOW to be a great issue as they are already walking in a race where the WOW is installed.

When dry stock use the WOW they will not be walking down races, they will have to go from a large area such as a field and funnel down into the narrow entry of the WOW. This could make it difficult to encourage them to walk across, unless they are attracted to something on the other side, such as feed (food) or sound of a tractor. The cattle cannot be forced over, as this will not result in good quality results, as found in a previous study (Teal).

The algorithm in the DairyScale hardware may need to be altered to serve the purpose of weighing dry stock cattle, because of the behavioural differences in these animals. It is expected that dry stock cattle will move over the WOW at a faster pace, compared to dairy cows.

The software checks the data for problems that could cause weight estimates to be inaccurate. It is assumed that some of these error checks would need to be modified for dry stock cattle.

One of the error checks of the DairyScale requires the animal to cross the weighing platform 12 times before it can predict a weight range which the estimated weight should fall into. If the estimated weight is outside of this range it gets disregarded. This error check is useful in a dairy application where the cows are weighed twice a day, but not for dry stock as they are only weighed as required.

2.3.3 Hardware

The new algorithm for weighing cattle must run on the DairyScale hardware. The DairyScale hardware is designed to be used for walk-over weighing of dairy cows. The hardware is not very powerful, so any new algorithm must not be too computational intensive.

No user input is required to capture weights, it is able to detect and record the animal's weight as it walks over the platforms.

The DairyScale is limited to reading the load cells at 41.667Hz because of the limitations of the hardware inside the scale head (DairyScale unit). The code is written in uC and runs on a M16 Renesas (Mitsubishi) processor (Gallagher Group Limited).

SmartReader (RFID tag reader) cannot be run in continuous mode, because the DairyScale cannot handle the input. In continuous mode the SmartReader would continue to send the EID tag number while it is within range of the reader. As the DairyScale cannot cope with that much data, the SmartReader is setup to send the tag only once when it is within range of the reader.

The setup being used for this project can only run for a limited time, as the system is on a trailer for portability. The time that the DairyScale equipment can run for is limited by the amount of energy the batteries can store, and the rate at which the solar panels can replenish them. When used for dairy cows it would be installed in the exit race of a milking shed, so it could run off mains power or internal batteries.

2.4 Conclusion

This chapter has discussed various available weighing systems and methods. It covered some of the background constraints of dry stock cattle and limitations of the hardware. The approach taken to adapt the existing hardware and software to suit the behaviour of dry stock cattle is covered in subsequent chapters.

3.0 Development / Design – Gathering Data and Analysis

A good understanding of the environment, quality and limitations of the data available are required to development a walk-over weighing algorithm. This chapter describes the equipment in more detail, the various data collection exercises that were performed. It also examines the data, and describes in detail the ways the actual weights might be determined from the data.

3.1 Equipment

To conduct the research several pieces of specialised equipment were required. All equipment was supplied by Gallagher's, including the crate which was specifically built for this project.

The equipment used during the project:

- Crate
    - load cells
    - solar panels
    - batteries
    - winch
    - antenna
- DairyScale (XDS5000)
    - weighing platform
    - scale head
    - power supply
    - MyScale pro software
- SmartReader
- PC – suitcase, inverter, fans
- Laptop

3.1.1 WOW Crate

The WOW (walk-over weighing) crate is the piece of equipment that the cattle walk through, that can record data about them. This portable weighing crate was built previously for this project (see Figure 4).

The bottom of the crate has two steel platforms, on short lead-on and a long main platform. These have a thick piece of non-slip rubber attached to the top of them. Under the platforms there are three load cells, two under the main platform.

The crate has wheels and a winch. The winch allows the crate to be lowered onto the ground, for when animals are crossing. It can be raised onto its wheels when the crate is needed to be moved. It has a detachable draw bar (tow bar) attachment, so it can be easily moved from place to place using a vehicle or quad bike.

Two solar panels are attached to the top of the crate, which are used to keep the two 12v batteries charged. This limits the length of time the equipment can be run continuously for. The batteries are used to supply power for the winch, PC, DairyScale, and SmartReader.

An antenna is attached to the side of the crate; it is used to read RFID tags as animals walk through the WOW. The RFID reader (SmartReader) attaches to the back of the antenna.

Figure 4: WOW crate (exit end).

3.1.2 Load cells

The bottom of the crate has two platforms which the animals walk over. The first is a short lead-on platform, and second is the main platform.

The lead-on platform has one load cell under it, which is positioned in front of the main platform (see Figure 5). The lead-on platform was used as a switch in the DairyScale product, to announce the next animal was coming onto the WOW. A cable runs from the lead-on load cell to the DairyScale unit via a junction box.

The main platform has two load cells under it. The first one is referred to as the leading and the second one as trailing load cell. The leading load cell is placed after the lead-on platform; this is where the animals start walking on. The trailing load cell is placed at the end of the main platform.

The wiring for the leading and trailing load cells are joined together in a junction box under the crate. From this junction box, one cable connects to the DairyScale. Another cable is used to connect the DairyScale to the lead-on platform (see Figure 5).

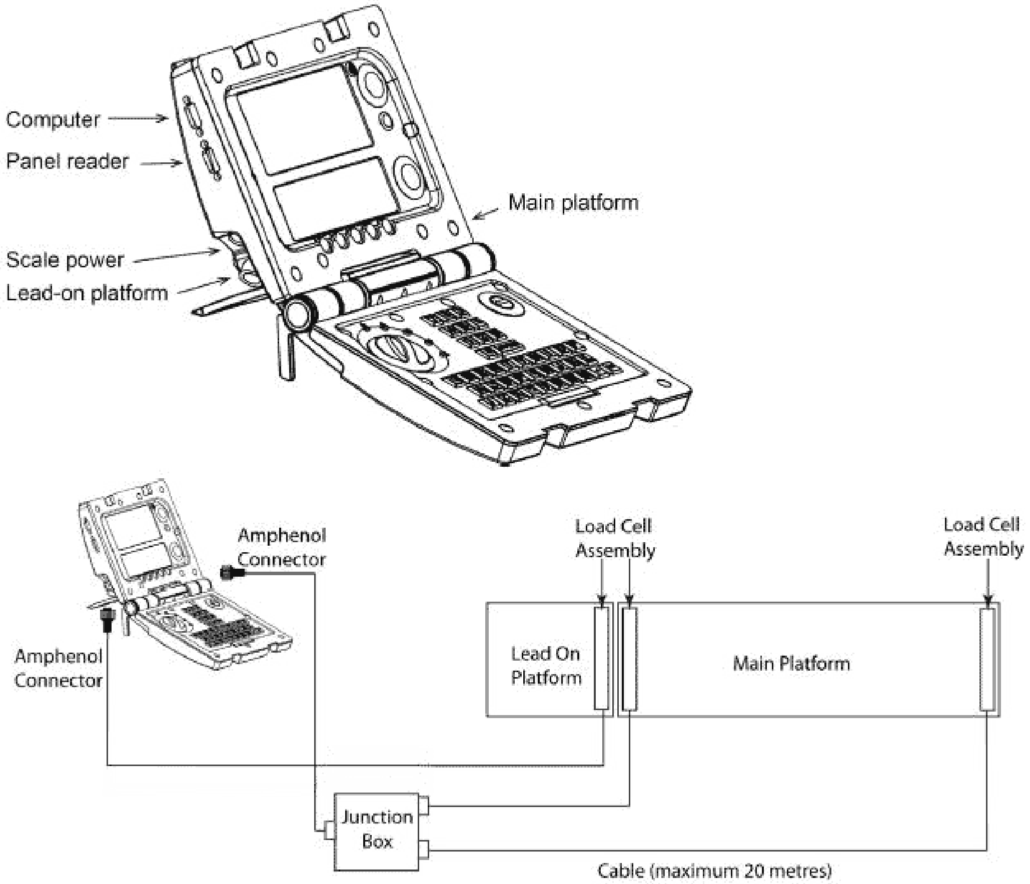

Figure 5: DairyScale scale head connected to three load cells.

(Gallagher Group Limited, 2011)

3.1.3 DairyScale Scale Head

The DairyScale unit (see Figure 6) is the central unit that connects to all the pieces of equipment in the WOW system.

The DairyScale unit can be used for static weighing as well as automatic walk over weighing (Gallagher Group Limited, 2011). The DairyScale has an inbuilt battery; it also can be plugged into the 12 volt power source that is available on the WOW trailer.

The DairyScale (scale head) unit served several roles for this research project:

- Producing estimated weights, using the DairyScale algorithm.

    - These estimated weights allow the weights estimated by new algorithm to be compared against them.
- Recording EID tag numbers, as they are read by the SmartReader.
- Outputting the raw load cell data, with EID tags directly to a capture application on an attached PC.
    - A hidden WeighDebug function must be activated to enable this output.

For this research static weights of the cattle were collected using a fixed static weighing system on the farm. As static weighing the cattle on the WOW would have been too difficult, because of the behaviour of the animals.

Figure 6: The DairyScale scale head.

3.1.4 SmartReader and Antenna

The WOW system is designed to weigh animals without supervision. Therefore a method of identifying animals as they are weighed is needed, allowing the weights to be associated to a particular animal.

New Zealand regulations now require that all bovines are tagged with an RFID chip that has an EID number for identification of the animal. This existing tagging system provided an opportunity to use the EID number as an identification of the animal to link it to the walk over events. The SmartReader and its antenna are attached to the side of the WOW crate, and are used to read the EID number in the RFID chips.

The WOW has an antenna panel (EID Tag Reader Antenna Panel 1300) attached to the crate, it plugs into the SmartReader (EID Tag Reader Controller R) unit (see Figure 7). The SmartReader requires a 12v power supply, provided by the batteries on the WOW crate. The SmartReader sends EID numbers to the DairyScale via a serial cable when they are read. The DairyScale matches the EID number to the appropriate weight estimation. The DairyScale also sends the EID's as part of the data it is continuously sending to an attached PC (when the output is set to WeighDebug).

All the cattle had RFID tags in their right ear; most of them were tagged with half duplex (HDX) tags. When the tag is in a range of an antenna, the SmartReader (EID controller) reads the unique number from the RFID chip in the tag.

Figure 7: SmartReader on the back of the antenna.

3.1.5 PC/Laptop

A laptop and PC were used for the project. The PC was used to collect data and the laptop was used to access the PC

Laptop

The laptop was used to view and control the PC, as the PC did not have a monitor, keyboard, or mouse. An application called UltraVNC was used. There is two parts to the application, a viewer and server. To access the PC with the laptop the UltraVNC viewer was used. A network cable is used to allow the two computers to communicate.

When using the UltraVNC viewer a computer name and a password is used gain access to the PC. This allows viewing and control of the PC, with the laptop.

Personal Computer

The personal computer (PC) is used to record the raw data from the load cells and SmartReader. A serial output from the DairyScale unit was configured to send a real-time data stream of each signal received by the DairyScale from the three load cells and any EID tags as they are read by the SmartReader. The raw data is recorded in CSV (comma separated values) format.

The PC was housed inside a plastic case (see figure 8), so it was protected from the farm environment while recording. There are two of fans around the PC in the plastic case, to prevent overheating. The PC is a DELL Inspiron zino hd PC, which requires a 19 volt DC power input. An inverter was used to convert the 12V supply from the car batteries to 19V. The PC has UltraVNC (server) installed, and is configured to automatically start up when Microsoft Windows starts up.

Figure 8: PC with inverter in plastic case.

3.2 Preparation

3.2.1 Data Collection Preparation

To confirm the WOW was ready to be used for weighing cattle the wiring needed to be checked. The load cell wiring can be checked by identifying if the load cell readings are within a certain range. This can be done by following instructions in the DairyScale's installer manual (Gallagher Group Limited, 2011). It was discovered that one of the load cells was wired incorrectly. The issue was rectified by following the wiring diagram in the installer manual.

Early in the preparation of the equipment it was noticed that some of the wiring was missing on the WOW crate. The data and power cables for the SmartReader (EID tag reader) had to be installed on the crate, in a way to prevent physical contact with the animals and therefore avoiding any possible damage or electric shock to the animal.

3.2.2 Test of Functionality

To test the RFID tags were being read, some EID tags through the WOW crate, by moving them past the antenna panel. The EID tags appeared in the raw data file which was being output to the PC, by the DairyScale.

3.2.3 Power System

The WOW trailer has two large deep cycle 12 volt batteries to store energy to run the devices. The batteries are charged by the two solar panels which are attached to the top of the WOW trailer.

The devices attached are the DairyScale, which has an inbuilt battery (and runs from 12v), SmartReader, and PC. Over a long period of time the solar panels were not able to supply the batteries with enough energy to keep everything running. The winch on the crate also uses these batteries.

3.2.4 Calibration

The DairyScale unit needs to be calibrated when it is attached to a set of load cells. A section in the DairyScale installer manual describes how to calibrate the load cells, so they can accurately measure weight. A total of 100kgs of test weights was used, in three different positions on the main platform.

3.2.5 Farm Selection

The farm where the trials took place had dry-stock cattle, which were tagged with RFID (EID) tag.

At the time of starting this research project the (NAIT) rules for mandatory RFID tagging were not in place. The options of farms with cattle that had RFID tags were slim. But after 1 July 2012 a regulations came into place which requires cattle to be tagged with RFID tags (NAIT Ltd).

The smooth running of the trials on the selected farm was possible thanks to a very helpful farmer. He spent many hours on several days helping with moving animals during the data collection sessions. He also did some preparation in his personal time to get the farm ready for the test equipment, such as modification of the stock yards.

3.3 Data Collection

3.3.1 Static Weighing

The dry stock cattle that were used on the farm had to be static weighed, then let into the stockyard. The static weighing was done in the farmers' permanent weighing crate. The data was recorded using a Gallagher TSi (scale head that was used for static weighing). The collected EID tag numbers and weights were easily downloaded onto a USB flash drive as a CSV file.

3.3.2 Stockyard Data Collection

The initial data collection was done in the farm's stockyard. This made it easy to static weigh the cattle and let them cross the WOW. The WOW was positioned at the opposite end of the pen to the static weighing equipment. Deer gates were setup on an angle, to funnel the bulls towards the entry of the WOW (see Figure 9).

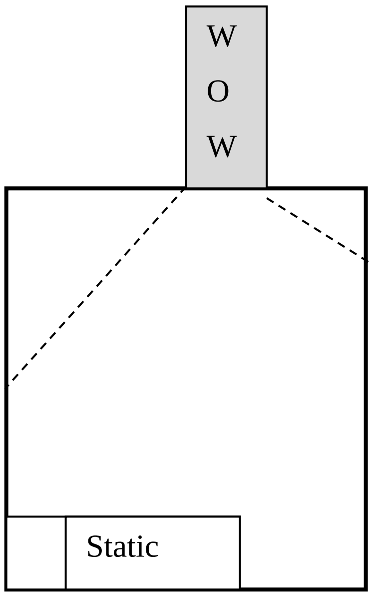

Figure 9: Stockyard weighing setup. The gates were setup (on an angle) to directed the cattle towards the WOW.

After setting up the WOW and a series of gates in the stockyard, the WOW was ready to have its first animals walk over it. The animals were wary at first. They initially had to be forced to go over the WOW, and they ran over it quickly which resulted in data that was difficult to use. However after the herd had been over the WOW several times they got familiar with the equipment, and walked over less hastily.

Understanding where the animal was physically positioned during its traversal of the WOW was difficult. The combination of fast moving animals and animals being too close to each other made the majority of the data unclear.

3.3.3 Data Collection with Video

It was decided that having a video of the cattle walking over the WOW would be useful, to see the exact position of the animal in contrast with the data. The WOW was setup with a video camera recording the side of the crate, so the hooves could be seen as the cattle walked across the platform.

Often when one animal crossed several would follow closely behind, resulting in two animals being on the platform at one time. This is a problem when estimating the weights of the animal because there can only be one animal on the WOW at a time, for the weight of the animal to have a weight estimated. To estimate the weight of an animal its full body weight must be on the main platform, without any other animal touching the main platform.

When animals are forced onto the WOW they do not behave in a natural way, which makes it hard to make use of the data for weight estimation.

3.3.4 Field Data Collection

It was decided that we should run the cattle over the WOW in the field, because we were hoping that the cattle would act in a more natural way. Also this situation would be encountered in some of the real life applications of the WOW device.

The farmer used a tractor and feed (silage) to attract the cattle to the WOW, by dropping some in the neighbouring paddock, so they had to walk over the WOW to get to the feed. This was repeated every morning for five days.

The animals walked/ran over the WOW much more calmly than they did in the stockyard. This was because they wanted to move across to the food being fed out by the tractor on the other side of the WOW. In contrast several of the cattle in the stockyards did not want to go through without being forced.

The data collection was setup to run for five days. Unfortunately when the data was analysed, a problem became apparent. The equipment ran for approximately 50 hours in very cloudy weather, so the solar panels did not manage to keep the batteries at an adequate voltage, cutting the data collection short. Subsequent data collections were performed over a shorter period of time with the researcher's presence. This allowed viewing of the walkover events that gave better understanding of the cattle behaviour and its relationship to the data.

3.3.5 Field Data Collection – Wider Range / Lighter Animals

It was decided to that a herd with a wider range of weights should be weighed, as the herd tested in the field previously was of a narrow range. A herd of 46 animals with a range of weights varying was lighter and had a range of weights from 183kg, up to 314kg were used. This varying weight data set would be a more comprehensive test for the devised algorithm. Unfortunately hardware problems, particularly in regards to EID tags were encountered and were unable to gain any useable data from the data collection.

3.3.6 Algorithm for Stockyard verses Field

The data collected in the field was clearer than the data collected from the animals in the stockyards. The animals spent a longer period of time crossing the WOW, so the distinctive features of the animals walk was easier to identify. The animals were more willing to walk across the WOW in the field, than in the stockyards, possibility because they knew there was food on the other side.

The proposal for the research project requires the application of the WOW to be suitable for use in paddocks (Harris, 2011). The DairyScale was designed to be used with placid animals walking out of a rotary milking shed, at a slow pace and doing it several times a day. Beef cattle are expected to be walking over the WOW at undetermined intervals, possibly several weeks or months apart.

3.4 Analysis

3.4.1 Copy Data from Equipment

The first item to download is the static data from farmers TSi (scale head). It is downloaded to a USB flash drive in CSV format.

The walkover data is recorded by the PC in a CSV formatted TXT file, using an application (supplied by Gallagher's). This file can be as large as 110Mb if it contains the maximum of 24 hours of data. These files become too large to edit in Microsoft Excel, so they can be split into shorter files. Splitting the large data files into one hour files was very useful.

To download the data from the DairyScale the MyScale Pro application from Gallagher's was used. The data from the DairyScale unit was sent over a serial to USB cable to the laptop computer.

The video was recorded on a digital video camera, with an internal flash memory. The video was downloaded using the supplied USB Mini B cable.

3.4.2 Matching DairyScale Estimated Weights to Static Weights

A spreadsheet had been created to automate the process of matching the DairyScale estimated weights to the static weights. The spreadsheet also has formulas which are used for calculating the accuracy of the DairyScale weight estimations. This spreadsheet was also used to match CSV application estimations to the true weights of the animals.

3.4.2.1 Invalid EID Tags

During data collections 2, 3, 4 several of the EID tags that were read by the SmartReader did not match up with the tags that were recorded during static weighing. It was discovered that some of the animals had two tags from different manufactures attached to their ears. The farm has been used for research previously, so that is a possible reason for the animals having two tags. The invalid tags had EIDs starting with number 951.

EID tags are formatted in this way: “942 000003055843”

All the tags which were recorded during static weighing started with 942.

951 is the Leader manufacturer’s code.
942 is the Zee manufacturer’s code.

The tags which we should be reading are the “Zee” manufactured tags, not the “Leader” manufactured tags.

3.4.3 Finding Net Weight

The DairyScale produces raw data as a gross weight for each load cell; this includes the weight of the platforms attached to the top of the load cells (Tare weight). The value of the tare weight needs to be calculated, so it can be removed from the gross weight of each data point before processing it. Only the Net weight is needed, as this is the weight of the animal.

This can be done by finding a section, usually at the start of the data collection which has no weight on the platform. Calculate an average over at least a second of data, for each load cell. These averages can be removed from each respective load cell, for every data point, to obtain the Net weight value. The load cells output in increments of 100 grams, so a value of 3000 (Net) corresponds to a 300kg animal.

During the data collection as animals walk over the WOW the weight of the platform will increase slightly, because the cattle will drag mud and stones onto the weighing platform.

3.4.4 Calculations on Data

By writing formulas in Microsoft Excel, it was possible to work out the start and end points of animal walk over events. By looking for sudden rises in the weight value.

Several different mathematical formulas were tested to find a method of estimating the weight of the animals. These or a combination of these were trialled; average, median, mode, count, standard deviation, minimum, maximum and range.

Weight calculations became successful when working with the moving window average of 40, which is explained below.

3.4.5 The Start and End Points

To locate the start of an animal crossing event the algorithm would look for a sudden rise in the value being output from the main weighing platform. Although this is not the most important start point to find, the important start point is when the entire weight of the animal is on the main platform, not just its front hooves.

The lead-on platform was not used to identify the start of an animal crossing event.

The DairyScale algorithm uses the lead-on platform as a switch, and is not used to calculate the weight of the animal. Its primary function was to identify if the next animal was about to step onto the main platform. This is so the second animals' weight would not be included in the first animals' weight. The DairyScale would ignore both animals, as it could not calculate either animal's weight accurately.

The end point can be identified, when the values of the main weighing platform drops and reaches zero, or a value near zero. Some allowance must be made for mud being trampled onto the platform.

This method for finding start and end points was later dropped for the threshold method.

3.4.6 Moving Window Average

Using a moving average window of 20 and 40 were tested on the data. A moving window average of 40 seemed to work well, because it is close to the length of time the full weight of the animal is on the main platform of the WOW. Forty data points is nearly equivalent to a period of one second, as there are 41.667 samples of data taken per second by the WOW. This moving window identified that averaging the period of time when the animal is fully on the platform, would result in a figure that corresponds to the animals' weight. The data within the window would be used for the final estimation of the animal's weight, and the data on either side could be ignored.

3.4.7 Threshold Technique

The threshold is used to find the start and end points of an animal crossing over the WOW platform.

The threshold is set at a weight which is higher than the activity which is caused by two hooves, but less than the full weight of the animal, in most cases. After the initial averaging of figures between the start and end points, the start and endpoints are moved up to the average which was calculated, and this is repeated twice.

The detailed instructions of how it can be performed in an Excel spreadsheet are below.

1. Calculate the Tare weight for the leading and trailing load cells, and remove them from their respective data columns.
    a. The Tare weigh can be calculated using the figures at a time when there is no activity on the WOW platform, usually at the start of the CSV file. This is done by averaging at least 20 data points.
    b. Remove 2800 (280kg threshold) from each load cell.
2. In a new column (threshold average) add the leading and trailing figures together, and divide by two.
3. The figures above zero for each (animal crossing) event can have an average applied to them.

a. In a new column start calculating the average, referred to as "average column". From the first cell (that is above zero) in the "threshold average" column.
b. Then for the next row the average of the first and second cell, the third row, the average of the first, second, and third. Continue this until the end of the positive values, in the "threshold average" column.

4. Use the last average from the "average column", for the specific crossing.
5. The next column is used to find the start point. Check if the figure in the "threshold average" column is greater than the average (from the bottom of the "average column"). Also check that the value in the "threshold average" column in the previous row is less than the average (from the bottom of the "average column"). When both of these conditions are true (the first time) this is the start point.
6. The next column is used to find the end point. Check if the figure in the "threshold average" column is less than the average (from the bottom of the "average column"). Also check that the value in the "threshold average" column in the previous row is greater than the average (from the bottom of the "average column"). When both of these conditions are true (furthest down the list) this is the end point.
7. Calculate the average from the figures in the "averages column" from the new start point to the new end point to find a new average.
8. Use this new average to replace the figure used from the "averages column" previously. Repeat steps 5 and 6 twice. This will bring the average up, closer to the middle of the activity which occurs when the animal is fully on the main platform.
9. Add 2800 to the new average (after the iterations).
10. Divide by 10, which results in the estimated weight in kilograms.

3.4.8 Threshold Technique – with Graphs

This section describes in detail the graphic interpretation of a dry stock animal crossing the walk-over weighing platform.

The threshold was set at 2800 for the spreadsheet on which the graphs were based; a straight thick black trace on the graph (Figure 10) shows the threshold. The threshold line appears horizontally above the activity caused by two hooves and below the activity caused by four hooves on the main platform.

The blue dot shows the point in time (x axis) which the EID tag was read. It appears along the 3000 line (y-axis), with no relation to weight.

The thin red trace in the graph shows activity on the lead-on platform. This uses a secondary y-axis scale which is not visible on the image; the magnitude is not significant to because it is not used to estimate weights.

The black trace is the average of the two load cells under the main platform, with the Tare weight removed.

This data was recorded on 02/08/12 (Appendix A – DVD, DC4). The location of the WOW was between two paddocks. The cattle were static weighed on the previous day.

Event 1

The thin red trace in Figure 10 represents the readings from the lead-on platform; it shows two rises before the animal is totally on the main platform. This is a common occurrence on the field data collected. The blue dot represents the time when the EID tag was read, in this case in the middle of the second red hump.

The first hump in thin red trace represents the front legs on the lead-on platform.

After the first hump of the red trace, the thick black trace rise indicates that the front legs of the cattle have moved onto the main platform (2000). The next rise of the black trace indicates further increase of the weight on the main platform, representing the animal now fully on the platform (about 3800).

The weight is taken off the lead-on platform (red trace plateaus). The weight comes off the main platform; this is shown by the drop of around half the weight (1500), when the front legs come off. When the rear legs come off the main platform, the weigh drops to near zero. The rise in the thin red trace near the end (right hand side) is the second animal stepping onto the lead-on platform.

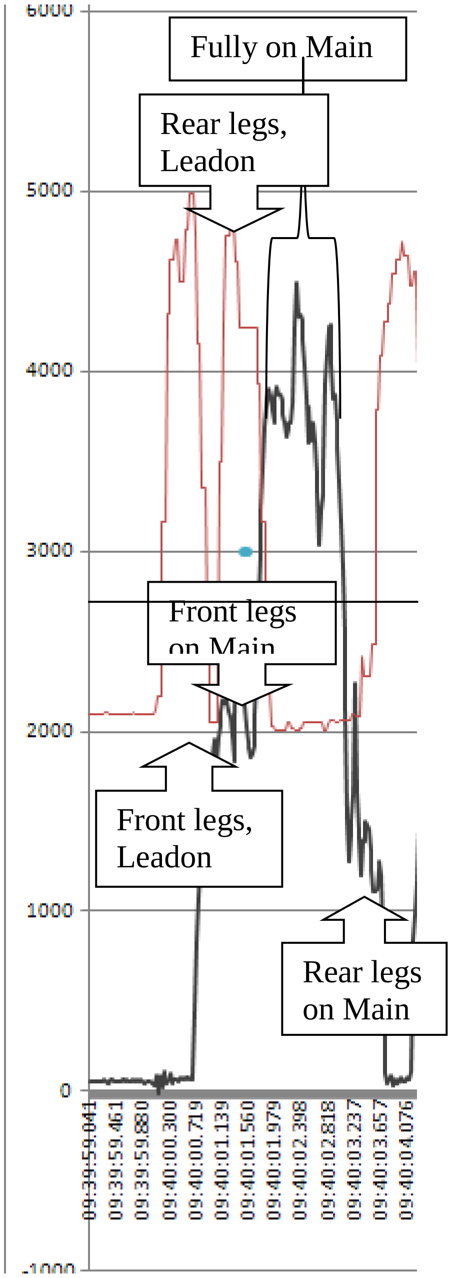

Figure 10: Event 1, describing features of an event.

Y-axis: Weight in 100 gram increments (1000=100kg).
X-axis: Time, hh:mm:ss:ms.
Blue dot: EID tag.
Thin red trace: Lead-on platform.
Thick black trace: Main platform.

Event 2

In Figure 11 a thick rectangle indicates the second animal crossing.

The first bump in the thin red trace is the front legs on the lead-on platform. The second bump is the rear legs on the lead on platform. At the same time there is a rise in the weight on the main platform (thick black trace) as the front legs transfer weight onto the main platform.

After the weight is off the lead-on platform, all the weight of the animal is on the main platform. (This is the time that calculations of the animal's weight are started)

At the time when there is a drop in the weight on the main platform (black trace), is when the animal's front hooves have stepped off the main platform. In this case (event 2) there is a rise in the black trace after this drop, because another animal is close behind. The thin red trace of the lead-on had a rise previous to this increase of weight on the main platform, which indicates that the next animal had stepped onto the lead-on platform.

The weight estimation was still able to be fairly accurate, because the weight on the main platform dropped below the threshold between the two animal events.

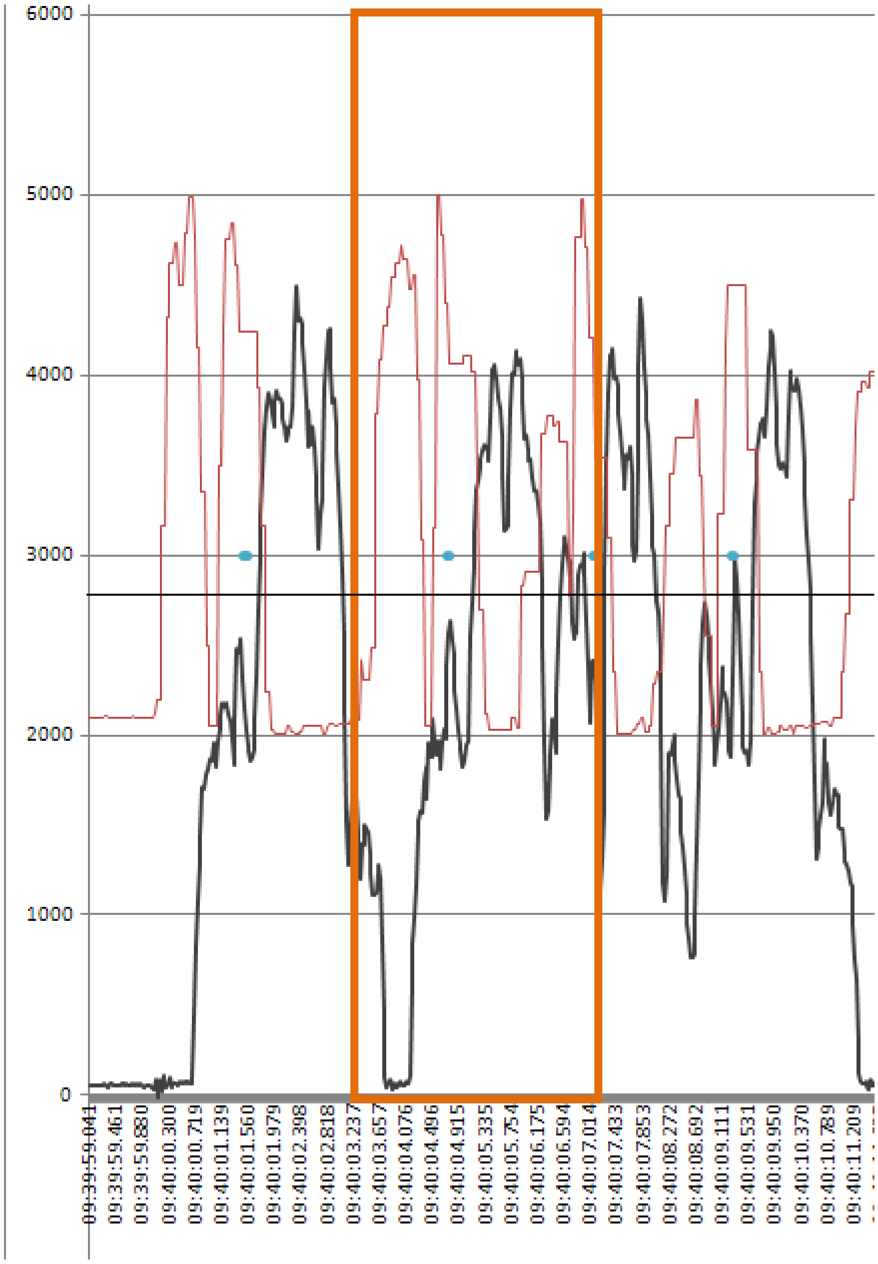

Figure 11: Event 1, 2, 3, 4

Y-axis: Weight in 100 gram increments (1000=100kg).
X-axis: Time, hh:mm:ss:ms.
Blue dot: EID tag.
Thin red trace: Lead-on platform.
Thick black trace: Main platform.

Event 8

These events 8, 9, and 10 in Figure 12 are examples of how the data looks when the cattle follow each other too closely on the WOW platform.

An automated method would find it difficult to locate a correct start and end points for all three events. The second animal stepped onto the main platform while the first one was still on it.

During the first two events the weight on the platform only dropped below the threshold briefly ($1/20^{th}$ of a second). It did not drop below the threshold between the two events, making it difficult to distinguish between the two animals'.

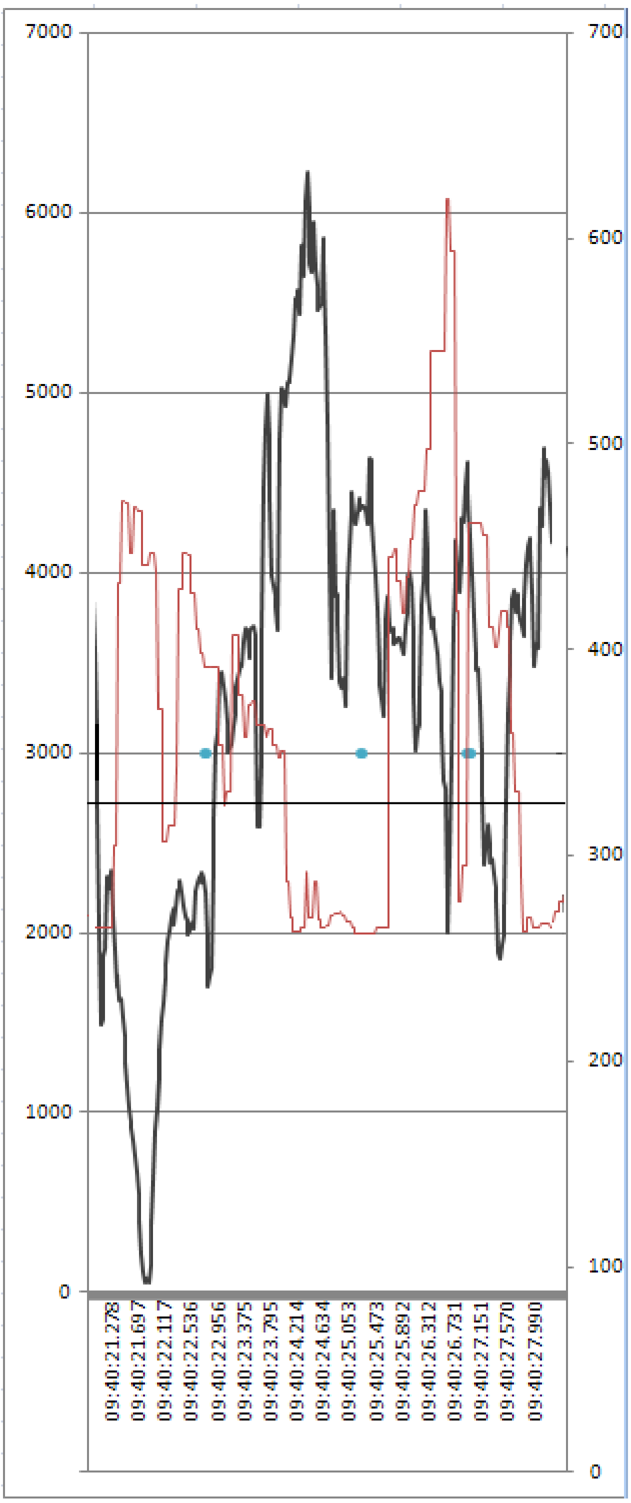

Figure 12: Event 8, 9, 10

Y-axis: Weight in 100 gram increments (1000=100kg).
X-axis: Time, hh:mm:ss:ms.
Blue dot: EID tag.
Thin red trace: Lead-on platform.
Thick black trace: Main platform.

3.4.9 Sum Threshold Technique

The sum threshold technique is a less computationally intensive method of performing the (initial design of the) threshold technique (see 3.4.7 Threshold Technique). This method is used in the CSV application (see 4.1 CSV Application) and is intended to be suitable for implementation on the DairyScale hardware. It uses subtraction and addition in conjunction with indexes for reference; this method replaces many of the averaging calculations, which were computationally intensive, used in the previous method.

The detailed explanations below are based on the spreadsheet shown in Figure 13 and 14.

	A	B	C	D	E	F	I	O	P	Q	R	S	T	U
15669	12:06:44.360	5	455	37436	35075		-80							
15670	12:06:44.389	4	455	37397	35158	3042953	-58							
15671	12:06:44.410	5	455	37337	35255	354kg	-39.5							
15672	12:06:44.434	5	455	37309	35305		-28.5							
15673	12:06:44.459	5	455	37285	35375		-5.5							
15674	12:06:44.484	5	455	37206	35476		5.5							
15675	12:06:44.508	5	455	37214	35525		34							
15676	12:06:44.533	5	455	37136	35579		22							
15677	12:06:44.558	5	455	37060	35623		6							
15678	12:06:44.582	5	455	36978	35638		-27.5							
15679	12:06:44.607	5	455	36680	35535		-228							
15680	12:06:44.632	5	455	36519	35534		-309							
15681	12:06:44.656	5	455	36489	35514		-334							
15682	12:06:44.681	5	455	36449	35471		-375.5							
15683	12:06:44.706	5	455	36375	35418		-439							
15684	12:06:44.730	5	465	36366	35408		-448.5							
15685	12:06:44.755	5	465	36371	35454		-423			Index				
15686	12:06:44.780	5	465	37042	35680		25.5	25.5		1				3360.5
15687	12:06:44.804	5	458	37450	35833		306	165.75		2	0	12501.5	0	
15688	12:06:44.829	5	458	37675	35831		417.5	249.667		3	0		0	
15689	12:06:44.854	5	427	37901	35967		598.5	336.875		4	0		0	
15690	12:06:44.878	5	427	38257	36066		826	434.7		5	0		0	
15691	12:06:44.903	5	427	38587	36122		1019	532.083		6	0		0	
15692	12:06:44.928	5	387	38758	36145		1116	615.5		7	0		0	
15693	12:06:44.952	5	387	38837	36168		1167	684.438		8	0		0	
15694	12:06:44.977	5	387	38852	36120		1150.5	736.222		9	0		0	
15695	12:06:45.002	5	387	38839	36061		1114.5	774.05		10	0		0	
15696	12:06:45.026	5	387	38805	36015		1074.5	801.364		11	0		0	
15697	12:06:45.051	5	279	38757	36001		1043.5	821.542		12	0		0	
15698	12:06:45.076	5	279	38770	36346		1222.5	852.385		13	0		0	
15699	12:06:45.100	5	279	38849	36663		1420.5	892.964		14	1		0	
15700	12:06:45.125	5	269	38925	36907		1580.5	938.8		15			0	
15701	12:06:45.150	5	269	38861	37370		1780	991.375		16			1	
15702	12:06:45.175	5	271	38694	37441		1732	1034.94		17				
15703	12:06:45.199	5	271	38596	37485		1705	1072.17		18				

Figure 13: Spreadsheet version of the sum threshold algorithm, part 1.

Column I:

- The average of both main platform load cells (Column D and E), minus the value of the Tare weight (See 3.4.3 Finding Net Weight) and the threshold (2181, usually 2800).
- Note: The threshold usually is 2800, not the 2181 that was used in this spreadsheet.

Column O:

- Is the average of the cells in column I (average starting at cell I15686)
- The second cell (O15687) in column O, averages the two cells I15686 and I15687
- The third cell repeats the process averaging the first three cells in column I, and so on. Until it reaches the last positive number in column I (I15778).
- The last average value (in cell O15778) is used to find the start and end point for the next iteration.

Column Q:

- Index number that is used for reference of position.

Column R:

- Top: Checks if the corresponding cell in column I is greater than the average from index 1 – 93 (O15778), and the preceding cell is less than the average. A value of 1 is placed in column R when this is true. Formula=IF(AND(I15687>O$15778,I15686<O$15778),1,0).
- Bottom: Checks if the corresponding cell in column I is less than the average from index 1 – 93 (O15778), and the preceding cell is greater than the average. A value of 1 is placed in column R when this is true. Formula: =IF(AND(I15778<O$15778,I15777>O$15778),1,0).

Column T:

- Top: Checks if the corresponding cell in column I is greater than the average from index 15 – 81 (S15782), and the preceding cell is less than the average. A value of 1 is placed in column T when this is true. Formula=IF(AND(I15687>S$15782,I15686<S$15782),1,0).
- Bottom: Checks if the corresponding cell in column I is less than the average from index 15 – 81 (S15782), and the preceding cell is greater than the average. A value of 1 is placed in column T when this is true. Formula: =IF(AND(I15778<S$15782,I15777>S$15782),1,0).

	A	B	C	D	E	F	I	O	P	Q	R	S	T	U
15760	12:06:46.606	5	271	35587	41140		2028	1342.32		75				
15761	12:06:46.630	5	271	35531	41111		1985.5	1350.78		76				
15762	12:06:46.655	5	271	35449	41147		1962.5	1358.73		77				
15763	12:06:46.680	5	271	35350	41083		1881	1365.42		78				
15764	12:06:46.704	5	271	35232	40817		1689	1369.52		79				
15765	12:06:46.729	5	271	35138	40617		1542	1371.68		80			1	
15766	12:06:46.754	5	271	35067	40375		1385.5	1371.85		81			0	2927.5
15767	12:06:46.778	5	271	34966	40117		1206	1369.82		82	1	10817	0	
15768	12:06:46.803	5	271	34932	40049		1155	1367.23		83	0		0	
15769	12:06:46.828	5	271	34936	39988		1126.5	1364.37		84	0		0	
15770	12:06:46.852	5	271	34881	39921		1065.5	1360.85		85	0		0	
15771	12:06:46.877	5	271	34975	39715		1009.5	1356.77		86	0		0	
15772	12:06:46.902	5	321	35077	40026		1216	1355.15		87	0		0	
15773	12:06:46.926	5	321	34840	39865		1017	1351.31		88	0		0	
15774	12:06:46.951	5	303	34967	39609		952.5	1346.83		89	0		0	
15775	12:06:46.976	5	303	34933	39418		840	1341.19		90	0		0	
15776	12:06:47.001	5	303	34925	39015		634.5	1333.43		91	0		0	
15777	12:06:47.025	5	303	34966	38551		423	1323.53		92	0		0	
15778	12:06:47.050	5	303	34974	38040		171.5	1311.15		93	0		0	
15779	12:06:47.075	5	446	34951	37377		-171.5						Total	104831.5
15780	12:06:47.099	5	446	35012	36155		-752				Total	111119.5	Count	64.0
15781	12:06:47.124	5	446	34931	36132		-804	Total	121937		Count	68.0	Avg.	1638.0
15782	12:06:47.149	5	468	34935	35964		-886	Count	93		Average	1634.1	TH	3819.0
15783	12:06:47.173	5	468	34916	35876		-939.5	Average	1311		Add threshold	3815.1	Est.	381.9
15784	12:06:47.198	5	484	34884	35902		-942.5				Weight est.	381.5	Diff.	27.9
15785	12:06:47.223	5	484	34909	35978		-892				Difference	27.5		
15786	12:06:47.247	5	484	34904	36745		-511							
15787	12:06:47.272	5	502	34809	37138		-362							
15788	12:06:47.297	5	502	34885	37306		-240							

Figure 14: Spreadsheet version of the sum threshold algorithm, part 2.

Column P:

- P15781: Total of column I, index 1-93.
- P15782: Count of the number of cells between index 1 and 93.

- P15783: Total (P15781) divided by count (P15782). Average, to be used for the next iteration.

Column S:

- S15687: A sum of the first 14 values in column I.
- S15767: A sum of the values between index 82 and 93 of column I is calculated.
- S15780: Total of column I (P15781) minus sum of the top 14 values in column I (total in cell S15687), and the sum of bottom values in column I index 82 – 93 (total in cell S15767). This calculates the total between indexes 15 and 81.
- S15781: Index 82 minus 14 (column Q). These cells were identified as being greater than the average (identified by the two TRUE values [1] in column R).
- S15782: Total (S15780) divided by count (S15781). Calculates the average.
- S15783: Average (S15782) plus the threshold value (Cell F6 = 2181), thus reintroducing the threshold.
- S15784: Divides the average (S15783) by 10. Estimated weight in kilograms.
- S15785: Estimated (S15784) minus the static weight. The difference between estimated and true weight.

Column U:

- U15686: The sum of the values in column I, from index 15 – 16 (I15700:I15701).
- U15766: The sum of the values in column I, from index 80 – 81 (I15765:I15766).
- U15779: The total from S15780 minus totals from U15686 (Figure 13) and U15766 (Figure 14). Calculates the total.

- U15780: Index 80 minus Index 16 (column Q). Calculates the count.
- U15781: Total (U15779) divided by the count (U15780). Calculation of the final average.
- U15782: Average (U15781) plus the threshold (cell F6 = 2181), thus reintroducing the threshold.
- U15783: Divides U15782 by 10. Weight estimated in kilograms.
- U15784: Estimated weight (U15783) minus the true weight. The difference in KG between estimated weight and true weight.

3.4.10 Recursive Calculation

When the animal crosses the platform the initial start point is identified by the weight passing upward through the threshold. As the animal walks over and off the platform the value output lowers, crossing through the threshold in a downward direction, this is the initial end point. The average can be calculated between these points. The average from the previous calculation is used as the new threshold. New start and end points are identified with this higher threshold. These points will span a shorter period of time, and a higher average figure will result, closer to the animals' weight. This recursive calculation is repeated (1, 2, or more times) until the previous average and the new average are the equal, which is when the estimated weight is identified.

3.4.11 Graph of a Normal Walk-over Event

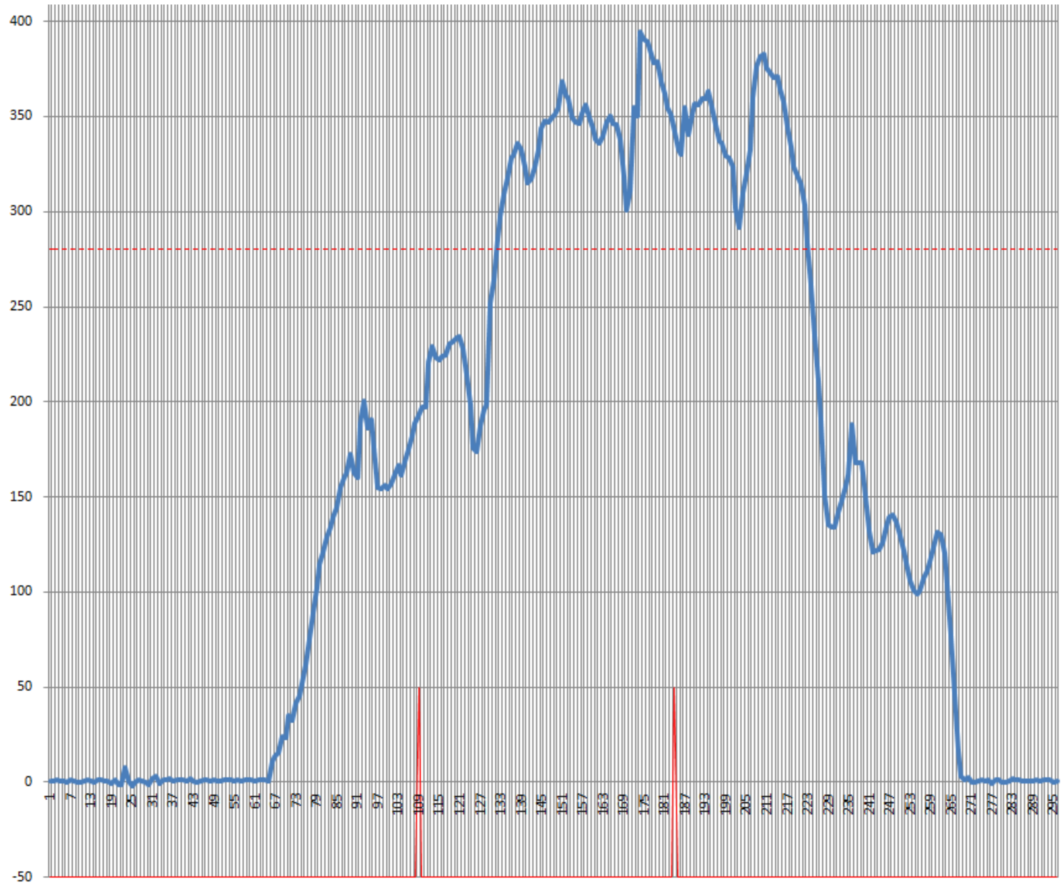

Figure 15: Graph of a typical walk-over event.

Figure 15 note:
EID 3035572 from 2/8/ 2012.
Y-axis: Kilograms.
X-axis: Time, at a rate of 41.667Hz.
Thick blue trace: Weight on main platform.
Dashed red line: Threshold (280kg).

This graph above (Figure 15) illustrates of an animal walking over the WOW. It shows a trace of both the leading and trailing signals of the main platform combined together. The two short spikes of the thin red trace in the bottom area of the graph indicate the point in time when the RFID tag was read. In this particular situation it shows an animal with two different RFID tags (only one tag was used, the other was excluded).

The dashed line indicates the threshold, set at 280kg (Y-axis). This can be adjusted in the code of the CSV application. During the project several different thresholds were experimented with. The threshold works best when it is greater than half the weight of the animal, as we only want to work with data which has the entire animals' weight on the main platform. A threshold of 280kg was selected as it proved to be the most suitable for a variety of herds which were weighed. Future development could include a menu option to alter the threshold, allowing the farmer to select a threshold appropriate for the herd.

The activity between 150 – 250kg on the upward movement of the weight, is displaying when only the front hooves are on the main platform. The activity between 100 – 200kg on the downward movement of the weight, is displaying when only the rear hooves are on the main platform.

4.0 Algorithm Development and Testing

This chapter discusses the weight estimation results and the various methods used to process the raw data.

The method used previously to process the raw data was a spreadsheet; it provided a quick way to trial several different calculations until a successful method was found. The problem with using the spreadsheet was that every event was individually processed, requiring a lot of time consuming interaction from the user. Another problem with the spreadsheet was a possibility of errors introduced by the user.

Another method of processing the data had to be developed to overcome the shortcomings of the spreadsheet method. A computer application was written for this purpose and named CSV (comma separated values) application, as it processes a CSV file that is produced by the DairyScale. The CSV application has minimal input from the user; it is unbiased and follows the rules of the algorithm, so it is consistent across all events. The algorithm is able to decide which events are good enough to be processed and decide how many iterations of averaging to perform for each event.

The chapter contain graphs of the results that illustrate the accuracy of each weight estimation method covered. Assessment of each weight estimation method is included.

4.1 CSV Application

The new algorithm used to estimate the weights of the dry stock cattle is called the threshold algorithm, as it uses a specified threshold to locate the start and end points of an animal crossing. The threshold is set at a level which is higher than the activity which is caused by two hooves, but less than the activity caused by all four hooves or the full weight of the animal.

The figures between the start and end points (grey circles in Figure 16) are averaged, then the average is used to locate new start and end points at a higher weight (stars in Figure 16), and this is repeated until the average stops moving up.

The sum threshold technique uses indexes and sums rather than "calculating averages repeatedly", as this is a less computationally intensive method of performing the calculations. Although this is not an issue on a modern computer, it would be an issue when running on the DairyScale hardware.

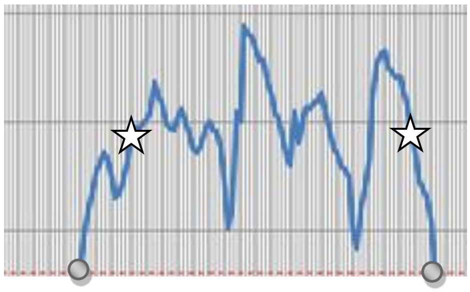

Figure 16: Part of a walk-over event, displaying the section above the threshold.

Figure 16 legend:
Grey circles: Initial start and end points.
Stars: Final start and end points.
Thick blue trace: Weight on main platform.

The CSV application was written to estimate weights from the WOW data using the devised sum threshold algorithm. The application processes the CSV output that was captured from the load cells of the WOW as an input. This captured data was taken at a rate of 41.667Hz; it contains raw weight data and the EID tags that were read.

The application is written in Realbasic, which has similar syntax to Visual Basic. The application can be compiled to run on Mac OS X, Microsoft Windows, or Linux.

4.2 Using the CSV Application to Test the New Algorithm

The software application was written to allow the new dry stock cattle algorithm to be executed with minimal human interaction. Previous to this the algorithm was applied to the raw data in an Excel spreadsheet. The sum threshold method was implemented in the code of the application.

There are three versions of the application. The first is the original version of the sum threshold algorithm (see Figure 17). The second has the same algorithm, but has an added button which allows the user to apply smoothing to the data previous to processing the data. The third makes sure that an event doesn't get cut short because of a drop below the threshold for a fraction of a second.

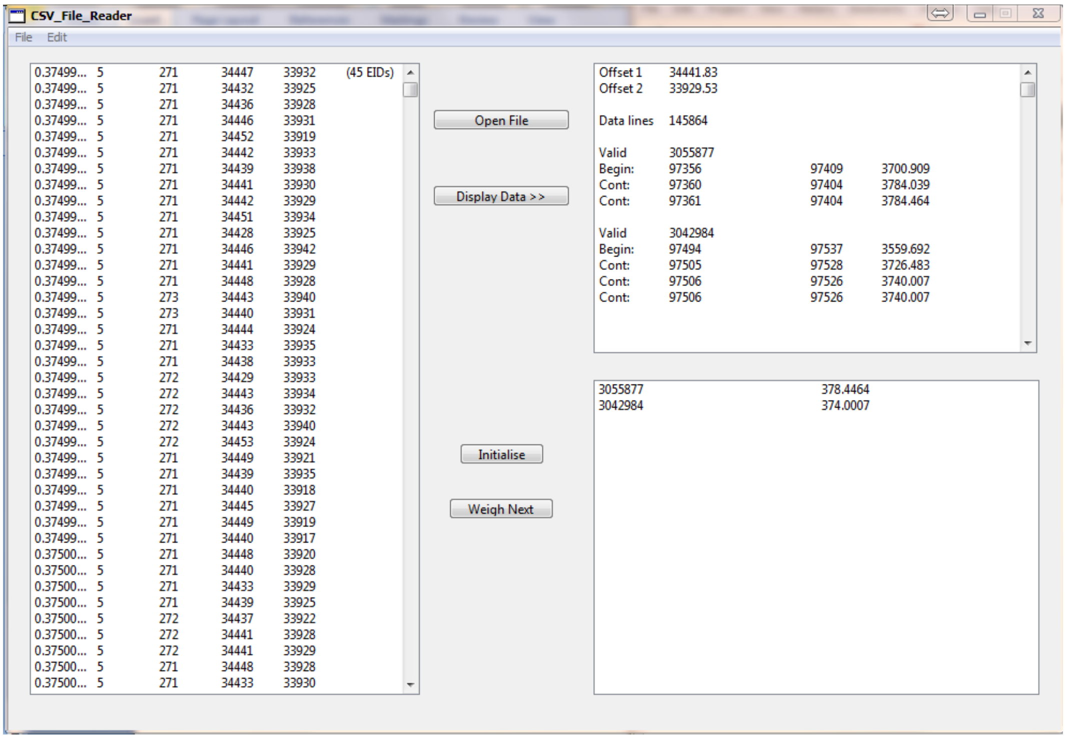

Figure 17: CSV application interface.

Application GUI:

- The left text area shows is a list of the input figures from the CSV (raw data from DairyScale).
- The top right text area displays several details about each animals crossing
  - Valid, is the EID tag of the animal crossing
  - Begin, is the initial calculation between the start and end points at the point that the readings cross the threshold
  - Each crossing has two or three "Cont", which is how many times the averaging calculation has been performed, after the original average calculation (from the threshold).
  - The figures on the far right displays the animals estimated weight (in tenths of a Kilogram)

- The text box in the bottom right (of Figure 17) displays a list of the EID tags (for two events) on the left and the estimated weights in kilograms on the right.

Application operation:

- Use the Open file button to select a file to process. Press “Initialise” to start the process.
- When a weigh has been estimated press “Weigh Next” to invoke the application to process the next crossing event.

Application input:

- The file must be of CSV format, but have a TXT extension at the end of the file name.
- The first column in the file is the timestamp, which needs to be converted into TIMEVALUE format. The CSV file from the DairyScale unit is not formatted correctly for the CSV application.

Application parameters:

In the raw data collected the EID tag was (nearly) always recorded earlier than the weight crossing through the threshold. In turn the CSV application checks that the ear tag is recorded before the upward crossing of the threshold.

At times the weight of the animal goes over the 2800 (280kg) threshold for a fraction of a second, but this is not enough information to estimate the weight of the animal with. The weight must stay above the threshold for more than 240ms, to make sure the period of the event is long enough to be usable. If the weight is above the threshold for less than 240ms, the application will look for the next rise above the threshold which lasts longer then 240ms to identify the start of the event.

The threshold is a variable in the code of the application, which could be adjusted if needed. If the code was on the DairyScale hardware this could be an option in the menu system. It would be useful for farmers wanting to weigh younger/lighter animals.

4.2.1 Skipping Invalid EID Tags

A range of invalid EID tags were included because some of the animals used had two different EID tags (see 3.4.2.1 Invalid EID Tags), in some of the data collections. Some of the animals had two RFID tags, and we only had static weights for one of the tags on each animal. The application disregards any tags which are outside of the three to four million range (3000000 to 4000000). This would not be required when processing other data sets, or in a product for the end user.

4.3 DairyScale and Algorithm Results

This section takes a detailed look at the accuracy of both weight estimation algorithms; DairyScale and the threshold algorithm devised for dry stock cattle.

The accuracy of the DairyScale algorithm (for cows) was to be able to estimate weights within 3% of the actual weight, for 80% of the results (Teal). This does not include the events that have not had weights estimated for them due to error checking removing them. This has been used as a guideline of accuracy for the dry stock algorithm.

Table 1 below displays a summary of results for the error distribution graphs. The data was collected on three different days, on two parts of the farm. The data was processed with two different algorithms. The first is the DairyScale algorithm, which was processed in real time on the DairyScale hardware. The second was processed using the new algorithm in the CSV application on a PC. Both sets of results were compared to the actual weight of the animals (static weight data). A frequency bar chart and error distribution histogram was created to display the accuracy of each data set.

Table 1: Summary of CSV application and DairyScale error distributions.

Algorithm	*No. of animals*	*% actually weighed*	*% within 3%*
A – DairyScale DC1	90 (76 EID)	(51) 57%	57%
B – CSV App DC1	90 (82 EID)	(51) 57%	16%
C – DairyScale DC3	90 (113 EID)	(29) 32%	70%
D – CSV App DC3	90 (149 EID)	(78) 87%	18% (60% at -7%)
E – DairyScale DC4	30 (45 EID)	(13) 43%	69%
F – CSV App DC4	30 (45 EID)	(29) 97%	31% (69% at +4%)

Number of animals and passes performed:

- DC1 refers to (DC1 Day 2, herd 4, stockyard). 45 animals (two passes).
- DC3 (Stockyard) 30 animals (three passes).
- DC4 (Field) 30 animals (one pass).

See Appendix A for raw source data.

See Appendix C – Weight Estimation Results for event results.

Detailed information and graphs are in section 4.3.3 DairyScale and CSV Application Estimations below.

4.3.1 Invalid Tags

During a few of the data collection sessions it was found that some of the animals were registering tags that had not been recorded during the static weighing. This resulted in the animals not been able to be compared to a static weight.

These invalid tags are from a different manufacturer, which can be identified by the leading digits of the EID number. The most likely reason for an invalid EID tag is that some of the animal had two tags. It was often observed that an invalid

EID tag was recorded preceding or following a good tag, within a very short time interval.

4.3.2 Frequency Histogram

Error distribution graphs are used to display the accuracy of the weight estimates for a particular data collection session. These graphs are used for all the different algorithms tested; the DairyScale algorithm and three CSV application variations.

The reason for using the frequency distribution is to be able to visually see how the error of each algorithm is distributed, and how many of the weight estimations fall within 3% of the animals true weight.

The DairyScale was designed to weigh dairy cows and be able to obtain an accuracy of 80% of the recoded events to fall within 3% of the animals' true weights (Teal). Likewise the required accuracy for the dry stock cattle project is to obtain 80% of the results within 3% of the true weight.

Axes:

- The y-axis is frequency.
- The x-axis is the percentage error; percentage is displayed as a decimal.

4.3.3 DairyScale and CSV Application Estimations

A - DairyScale Estimations DC1

The herd of 46 animals was run though the WOW twice, but one of the animals did not have a static weight recorded. The DairyScale was not able to be estimate weights for all of the walk-over events, 25 of them had been given 'zero' weight. One of the main reasons for this is the error checking determined that the event had a high chance of the weight estimation being inaccurate. This is often caused by animals walking to close to each other on the weighing platform.

The DairyScale was able to estimate weights for 51 of the walk-over events, with valid EID tags. Of the 51 weight estimates, 29 or 57% are within 3% of the actual weight of the animal. The algorithm was able to actually estimate weights for 57% of the animals that crossed the WOW.

The accuracy of the DairyScale is expected to be able to have 80% of the recoded events, within 3% of the static weight. The DairyScale algorithm is designed for dairy cows, so it was not expected to work as well with beef cattle. It was useful to test the DairyScale algorithm, so the new algorithm can be compared against it.

Table 2: Error frequency distribution of DairyScale estimates vs. static weight for DC1.

Bin	*Frequency*	*Percentage*
-0.1	2	4%
-0.05	2	4%
-0.04	4	8%
-0.03	0	0%
-0.02	5	10%
-0.01	11	22%
0	7	14%
0.01	4	8%
0.02	1	2%
0.03	1	2%
0.04	2	4%
0.05	3	6%
0.1	3	6%
0.2	2	4%
0.3	0	0%
0.5	4	8%
More	0	0%
Total	51	

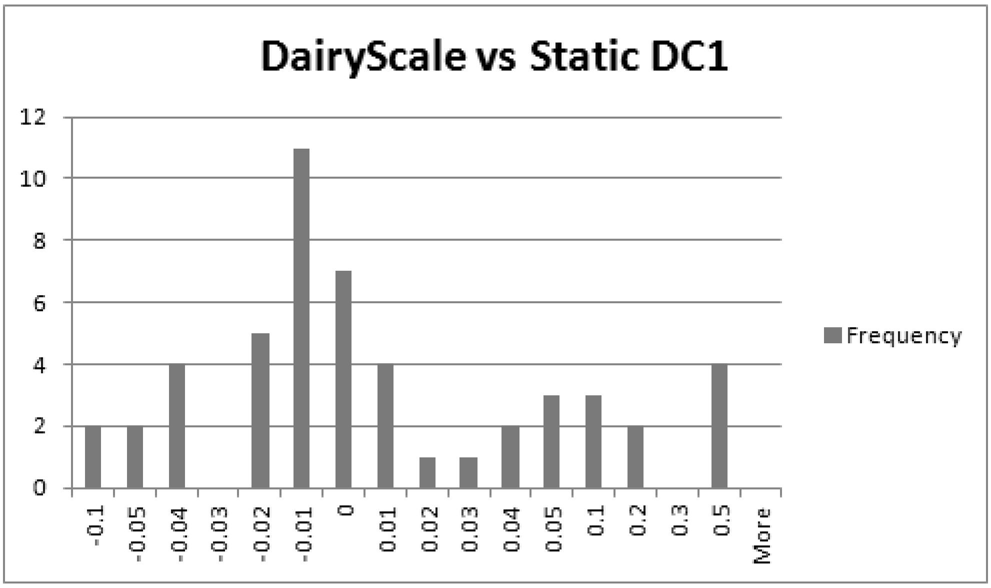

Figure 18: Error frequency histogram for DairyScale estimates vs. static weights of DC1.

B - CSV Application Estimations DC1

The herd of 46 animals was run though the WOW twice, although one animal did not have a static weight recorded. The raw data file contained 82 EID's, 22 of these were invalid and nine were unaccounted for. The CSV application estimated 51 weights with the sum threshold algorithm, resulting in 57% actually getting weighed. The accuracy achieved by the threshold algorithm for this data set was 8 (16%) of the 51 valid events were within 3%. The majority of the weights were overestimated by the threshold algorithm.

Table 3: Error frequency distribution of CSV application estimates vs. static weight for DC1.

Bin	*Frequency*	*Percentage*
-0.3	6	12%
-0.2	5	10%
-0.1	0	0%
-0.08	0	0%
-0.06	0	0%
-0.04	0	0%
-0.03	0	0%
-0.02	1	2%

0	0	0%
0.02	3	6%
0.03	4	8%
0.04	2	4%
0.06	8	16%
0.08	9	18%
0.1	5	10%
0.2	8	16%
0.3	0	0%
More	0	0%
Total	51	

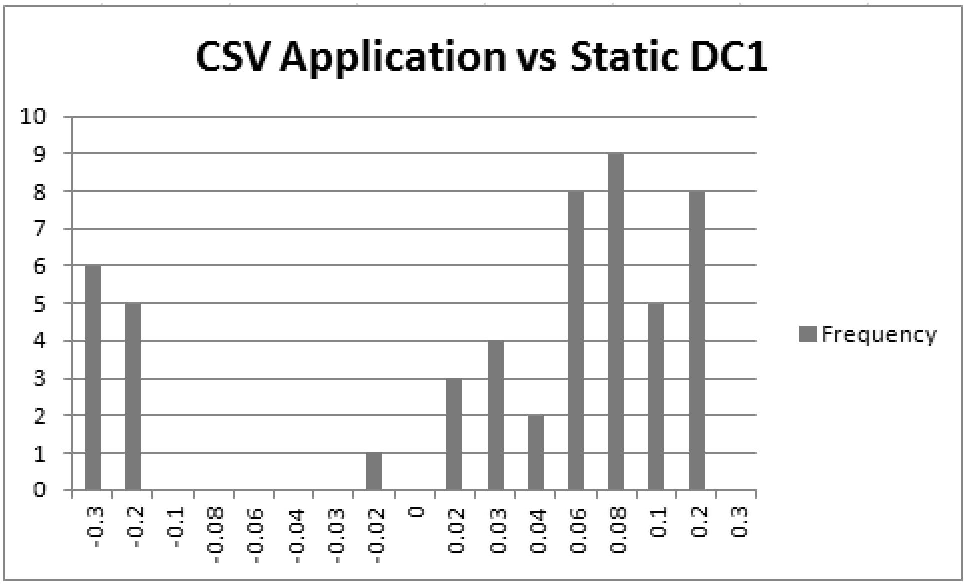

Figure 19: Error frequency histogram for CSV application estimates vs. static weights of DC1.

Stockyard DC1 Conclusion

The error checking of the DairyScale algorithm dropped about a third of the animals. The most likely reason for this is that the animals were walking to close to each other. The algorithm was able to estimate over half of their weights within 3% of the true weight.

Both DairyScale and CSV application estimated weights for the same number of animals. The threshold algorithm was only able to achieve 16% of the weights being within 3% of the actual weight. The achieved accuracy of the threshold algorithm was poor, as it had a tenancy to overestimate weights with this data set.

C - DairyScale Estimations DC2

The data collection was performed using the WOW in the farms stockyards with a herd of 30 animals. They were run though the WOW three times. The DairyScale algorithm estimated 46 weights, of which 29 of the weights estimated had valid EID tags (see 3.4.2.1 Invalid EID Tags).

The accuracy of the DairyScale with dry stock cattle is near the desired 80%, with 70% of the DairyScale's estimations being within 3% of the animals' true weight. Although only 32% of the animals that walked over had weights estimated for them, which is very low.

Table 4: Error frequency distribution of DairyScale estimates vs. static weight for DC2.

Bin	*Frequency*	*Percentage*
-0.2	1	3%
-0.1	2	7%
-0.05	1	3%
-0.03	1	3%
-0.02	0	0%
-0.01	4	14%
0	9	31%
0.01	3	10%
0.02	4	14%
0.03	0	0%
0.05	2	7%
0.1	1	3%
0.2	1	3%
More	0	0%
Total	29	

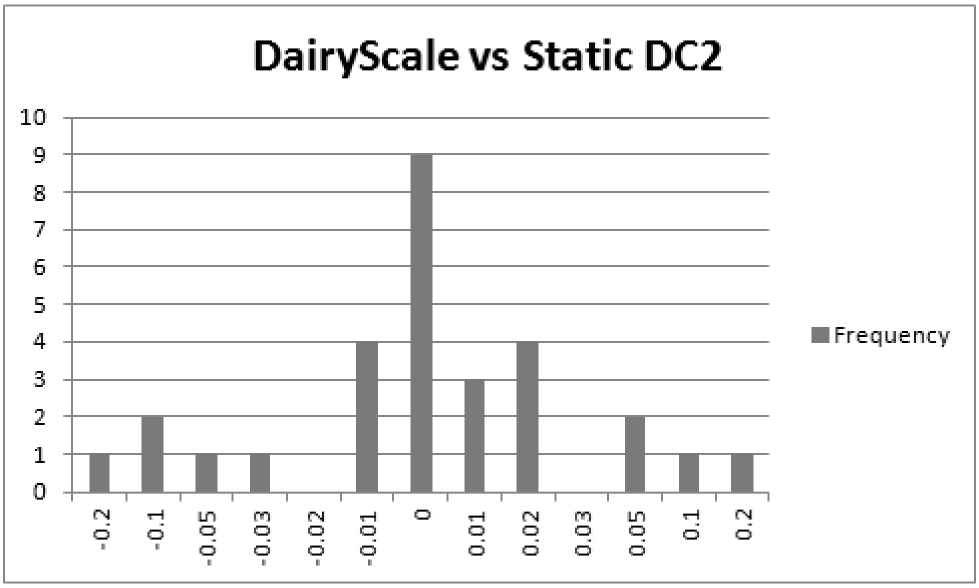

Figure 20: Error frequency histogram for DairyScale estimate vs. static weight of DC2.

D - CSV Application Estimations DC2

A herd of 30 cattle crossed the WOW three times during this weighing session in the stockyards. The raw data was processed by the CSV application using the (sum) threshold algorithm. It estimated weights for 78 animals. Although a poor accuracy result of 18% within 3% of true weight was achieved. The majority of the results were underestimated by the algorithm.

If a correction factor of -7% was to be used, 60% would be within 3%. The grouping of results was not ideal. This is an example of a bad set of results from the threshold algorithm.

Table 5: Error frequency distribution of CSV application vs. static weight For DC2.

Bin	*Frequency*	*Percentage*
-0.5	3	4%
-0.3	7	9%
-0.2	4	5%
-0.1	11	14%
-0.09	6	8%
-0.08	4	5%

-0.07	4	5%
-0.06	3	4%
-0.05	9	12%
-0.04	10	13%
-0.03	9	12%
-0.02	3	4%
-0.01	0	0%
0	1	1%
0.01	0	0%
0.02	1	1%
0.03	0	0%
0.05	0	0%
0.1	1	1%
0.2	1	1%
0.3	0	0%
0.5	1	1%
More	0	0%
Total	78	

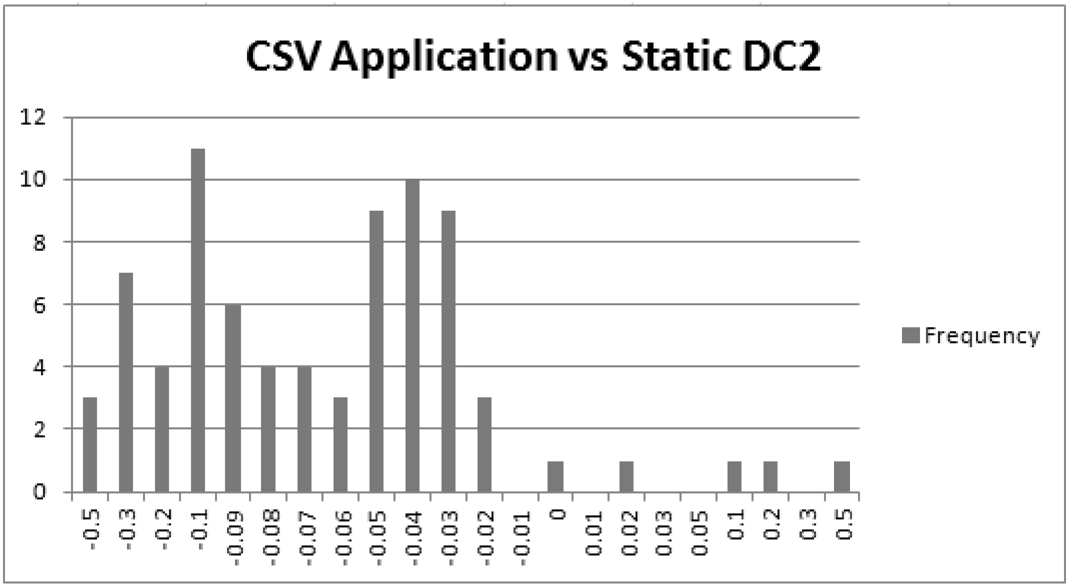

Figure 21: Error frequency histogram for CSV application vs. static weight of DC2.

Stockyard DC2 Conclusion

DairyScale:

Accuracy of the resultant DairyScale weights was good. The DairyScale was designed to get 80% of the estimated weights within 3% for dairy cows, this data collection with dry stock cattle achieved 70%.

Only 32% of the animals that crossed the WOW had weights estimated for them. This low rate of weighing estimates is caused by the error checking built into the DairyScale algorithm. The positive side of this is that the weights that were estimated are more accurate. The error checking was able to discard events that would result in inaccurate results.

CSV:

The CSV application attempted to estimate weights for a larger proportion of the animals, 87% of the animals. The algorithm was not able to achieve an accuracy that was acceptable from this stockyard data collection. Only 18% of the results were within 3% of the true animal weights.

E - DairyScale Estimations DC3

Static weights for the herd were recorded in the morning of the previous day (1/8/12). The data collection was performed between two paddocks from midnight until 10:40am on 2/8/12.

Of the 30 animals that crossed the WOW 16 weights were estimated, but 3 of the EID tags were invalid. This resulted in 13 or 43% with valid EID tags actually being weighed. The accuracy of the DairyScale algorithm for this data set was 69% of the results being within 3% of the true weight.

Table 6: Error frequency distribution of DairyScale estimates vs. static weights for DC3.

Bin	*Frequency*	*Percentage*
-0.1	2	15%
-0.08	0	0%
-0.06	1	8%

-0.03	1	8%
-0.02	1	8%
-0.01	1	8%
0	1	8%
0.01	1	8%
0.02	3	23%
0.03	1	8%
0.06	1	8%
0.08	0	0%
0.1	0	0%
More	0	0%
Total	13	

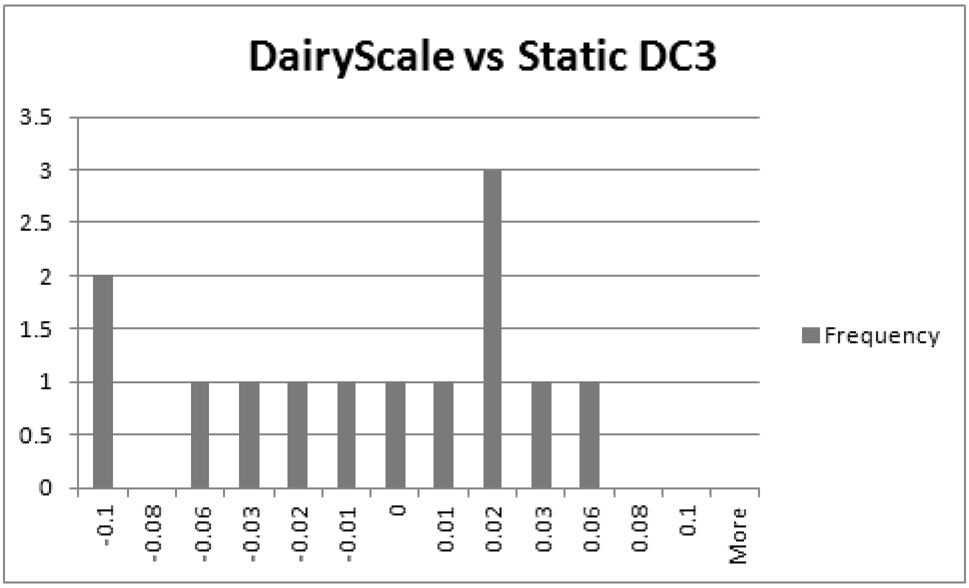

Figure 22: Error frequency histogram of DairyScale estimate vs. static weights of DC3.

F - CSV Application Estimations DC3

A herd of 30 cattle were static weighed on the previous day to the data collection (morning of 1/08/12). The WOW data was recorded in between two paddocks using the WOW crate between 9am – 10am of 2/08/12.

During the crossings of the WOW 45 EID tags were recorded, of which 15 were invalid EIDs. The algorithm estimated weights for 29 of the 30 valid EID readings

(97% actually weighed). The reason for one of the 30 tags not getting a weight estimated for it was that the animal was too far across the platform when the EID tag was read.

The results of the CSV application initially do not look good with only 31% of the estimated weights being within 3%. If a correction factor of +4% is applied to allow for weight the animals have gained since static weighing then 69% of the results would be within 3% of the static weights.

The over estimation of weight may have been caused by food (feed) they had consumed between the time of static weighing and the WOW weighing session. The static weighing was performed approximately 22 hours before the animals walked over the WOW for this data collection.

Table 7: Error frequency distribution of CSV application estimate vs. Static weight for DC3.

Bin	*Frequency*	*Percentage*
-0.3	0	0%
-0.2	0	0%
-0.1	0	0%
-0.08	1	3%
-0.06	0	0%
-0.04	2	7%
-0.03	0	0%
-0.02	0	0%
0	2	7%
0.01	1	3%
0.02	2	7%
0.03	4	14%
0.04	7	24%
0.05	1	3%
0.06	2	7%
0.07	3	10%
0.08	0	0%

0.1	0	0%
0.2	3	10%
0.3	1	3%
More	0	0%
Total	29	

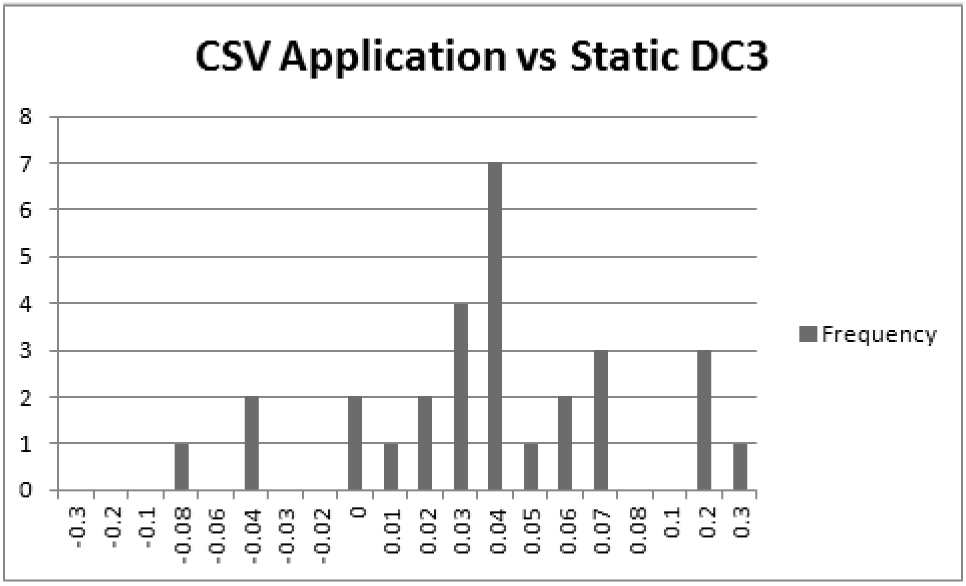

Figure 23: Error frequency histogram of CSV application estimates vs. Static weight for DC3.

Field DC3 Conclusion

The herd used for this data set had static weights recorded on the day before the data was collected. This was not an ideal situation, as the weights could have changed within the 22 hours between static weighing and the WOW data collection.

The WOW equipment was setup between two paddocks. The animals would walk through the WOW to get to the other larger paddock, to get to food (feed) the farmer had placed there. This motivated the cattle to walk over the WOW as they knew there was food on the other side.

The DairyScale algorithm worked well in the field. The accuracy of the DairyScale algorithm for this data set was 69% of the results being within 3% of the animals' true weight. Some of the animals did not get weights estimated for them, because of the error checking in the DairyScale algorithm.

The CSV application estimated weights for 97% of the animals that crossed the WOW, although only 31% were accurate to within 3% of the animals' true weight. The algorithm could have overestimated the weights slightly, or the cattle could have gained weight from eating.

DairyScale and CSV Application Estimations Conclusion

CSV application (sum threshold algorithm) attempts to estimate weights for a larger percentage of the animals, compared to the DairyScale. The error checking of the CSV application is not as strict as the DairyScale's, therefore allows the application to make estimations for more of the events. The down side is that it is not able to be as accurate.

The DairyScale does a good job error checking, especially when tracking location of each animal's hooves. It does this to detect multiple animals on the platform at once. This did result in many of the walkover events being dropped when animals walked too close to each other. This resulted in the estimated weights being more accurate than the CSV application using the (sum) threshold algorithm.

4.4 Revisions of the Algorithm

After the initial testing of the CSV application, two alternative versions were created to test two techniques which could possibly improve the results. One version had a button which allows the user to apply smoothing to the data before processing it. The second version was created to rectify a problem that had occurred occasionally. This problem occurred only when the weight dropped below the threshold for a fraction of a second, rising soon after. This was causing the algorithm to cut the event off too early, resulting in the weight being significantly overestimated.

This section below takes an in-depth look at the results from the revised versions of the CSV application.

The weight results can be found in the Appendix, under Data DC1 – CSV with Additions DC1 and Data DC3 – CSV with Additions DC3.

4.4.1 Pre-Smoothing

A suggestion that smoothing the data before processing it could improve results was made. A button was added to the CSV application, to allow the user to smooth the raw data before starting the calculations. A moving window average of 10 is run on both the leading and trailing load bar data columns. After the smoothing is applied the user can initialise the calculations to estimate the weights.

CSV with Pre-smoothing DC1

The shortened raw data file used to test the pre-smoothing addition contained 20 EID tags in it. The original CSV algorithm estimated weights for 13 of these events. After applying smoothing to the data the threshold algorithm estimated weights for eight of the 20 events.

From these estimations two were closer to the actual weight than the original threshold algorithm. Unfortunately the smoothing had a negative outcome on six out of the 20 events, and did not estimate weights for five events.

Table 8: Comparison of CSV with pre-smoothing to original CSV algorithm using DC1 data.

Bin	*Frequency (Smoothing)*		*Frequency (Original)*	
-0.3	1	13%	0	0%
-0.2	3	38%	1	8%
-0.1	1	13%	0	0%
-0.08	0	0%	0	0%
-0.06	0	0%	0	0%
-0.04	0	0%	0	0%
-0.03	0	0%	1	8%
-0.02	1	13%	0	0%
0	0	0%	0	0%
0.02	1	13%	2	15%

0.03	0	0%	2	15%
0.04	0	0%	2	15%
0.05	1	13%	0	0%
0.06	0	0%	1	8%
0.08	0	0%	1	8%
0.1	0	0%	2	15%
0.2	0	0%	1	8%
0.3	0	0%	0	0%
More	0	0%	0	0%
Total	8	100%	13	100%
Within 3%	2	25%	5	38%

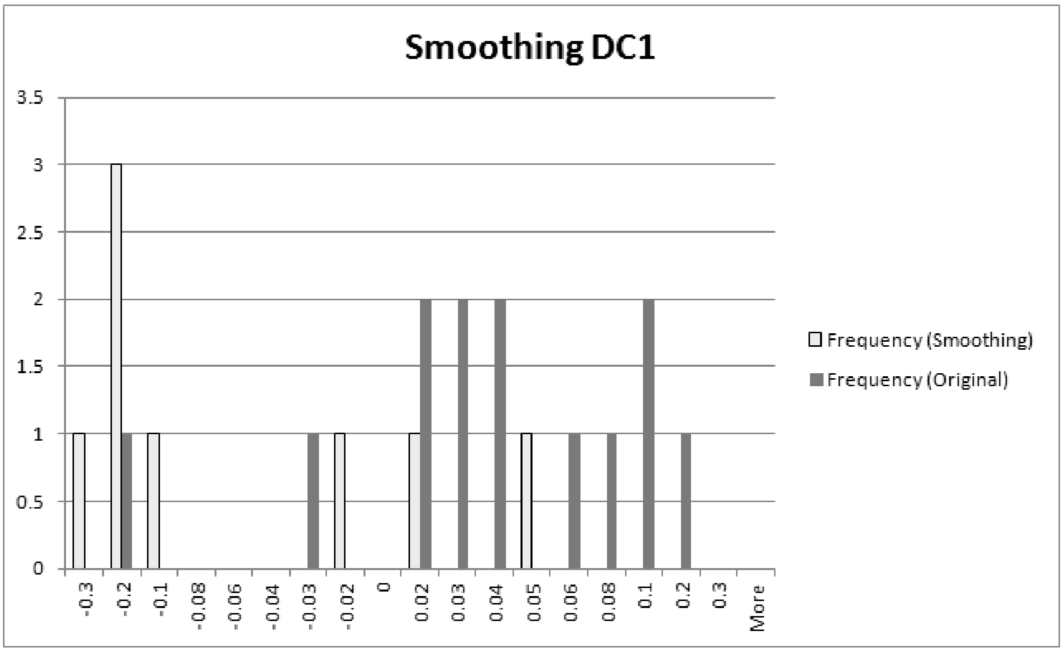

Figure 24: Comparison of CSV with pre-smoothing to original CSV algorithm using DC1 data.

CSV with Pre-smoothing DC3 (Field)

The pre-smoothing addition for the CSV algorithm produced results that caused the accuracy to drop significantly. The pre-smoothing made 13 events worse than the original threshold algorithm. Only one of the events was overestimated, the rest were noticeably underestimated. Only 13% (2) events were within 3% of the actual weight, whereas the original threshold algorithm had a better accuracy of 30% within 3% of the actual weight.

One of the results was improved by the smoothing. It was underestimated by only 0.7kg (0.2%), previously the original CSV (and premature finish) were both about 15% off.

Table 9: Comparison of CSV with pre-smoothing to original CSV algorithm using DC3 data.

Bin	*Smoothing Frequency*		*Original Frequency*	
-0.3	2	13%	1	3%
-0.2	2	13%	0	0%
-0.1	6	40%	0	0%
-0.08	1	7%	1	3%
-0.06	1	7%	0	0%
-0.04	0	0%	2	7%
-0.03	1	7%	0	0%
0	1	7%	2	7%
0.03	0	0%	7	23%
0.04	0	0%	7	23%
0.06	0	0%	3	10%
0.08	0	0%	3	10%
0.1	0	0%	0	0%
0.2	1	7%	3	10%
0.3	0	0%	1	3%
More	0	0%	0	0%
Total	15	100%	30	100%
Within 3%	2	13%	9	30%

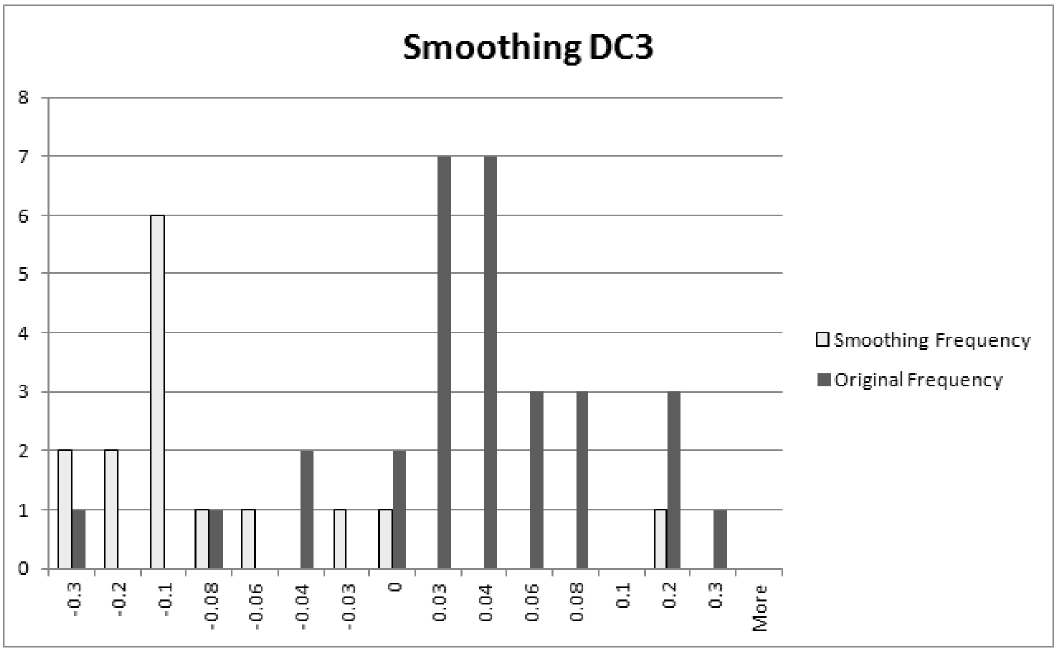

Figure 25: Comparison of CSV with pre-smoothing to original CSV algorithm using DC3 data.

4.4.2 Premature Finish

The reason for the premature finish addition to the CSV application was to avoid cutting events off too early. This is when the weight drops below the threshold only to raise a fraction of a second later. The normal algorithm was cutting the event off at this point, causing weights to be significantly overestimated (20%).

The new premature finish addition to the algorithm adds a check when the weight drops below the threshold, it looks for a rise above the threshold within the following 240ms (10 readings). If it does rise then it looks for the next drop below the threshold, to locate the true end point of the walk-over event.

The premature finish addition should only make a difference to events which dropped below the threshold before the end of the walkover event. The results of events that do not require this feature should have the same estimated weights as the original version of the CSV application produced.

The diagram below (Figure 26) shows an event that is an example of the weight dropping below the threshold for a fraction of a second. The weight dropped below the threshold at the 77th position (x-axis), rising up through the threshold at the 82nd position.

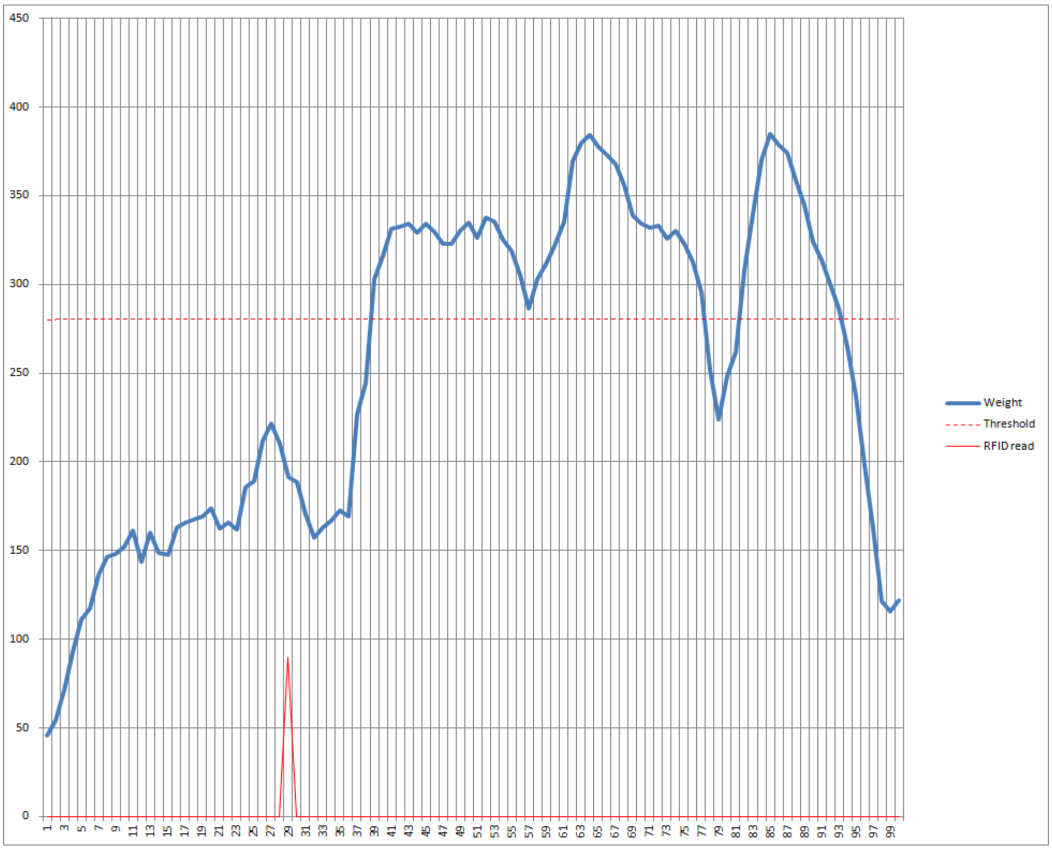

Figure 26: Example of an event where the weight dropped below the threshold prematurely.

Figure 26 note:
X-axis: Time at a rate of 41.667Hz per second.
Y-axis: Weight in KG's.
Thick blue trace: Weight on main platform.
Thin red trace: RFID tag read.
Dashed red line: Threshold (280kg).

CSV Premature Finish DC1

The premature finish addition estimated weights for 11 of the 20 walkover events, in the shortened raw data file. The results for 10 of the estimated weights were the same as the original threshold algorithm. The premature finish addition did not help improve this data set, as it missed two events and made one event worse.

Table 10: Comparison of CSV premature finish to original CSV algorithm using DC1 data.

Bin	*Frequency (PreM)*		*Frequency (Original)*	
-0.3	1	9%	0	0%
-0.2	1	9%	1	8%
-0.1	0	0%	0	0%
-0.08	0	0%	0	0%
-0.06	0	0%	0	0%
-0.04	0	0%	0	0%
-0.03	1	9%	1	8%
-0.02	0	0%	0	0%
0	0	0%	0	0%
0.02	1	9%	2	15%
0.03	2	18%	2	15%
0.04	2	18%	2	15%
0.05	0	0%	0	0%
0.06	1	9%	1	8%
0.08	0	0%	1	8%
0.1	1	9%	2	15%
0.2	1	9%	1	8%
0.3	0	0%	0	0%
More	0	0%	0	0%
Total	11	100%	13	100%
Within 3%	4	36%	5	38%

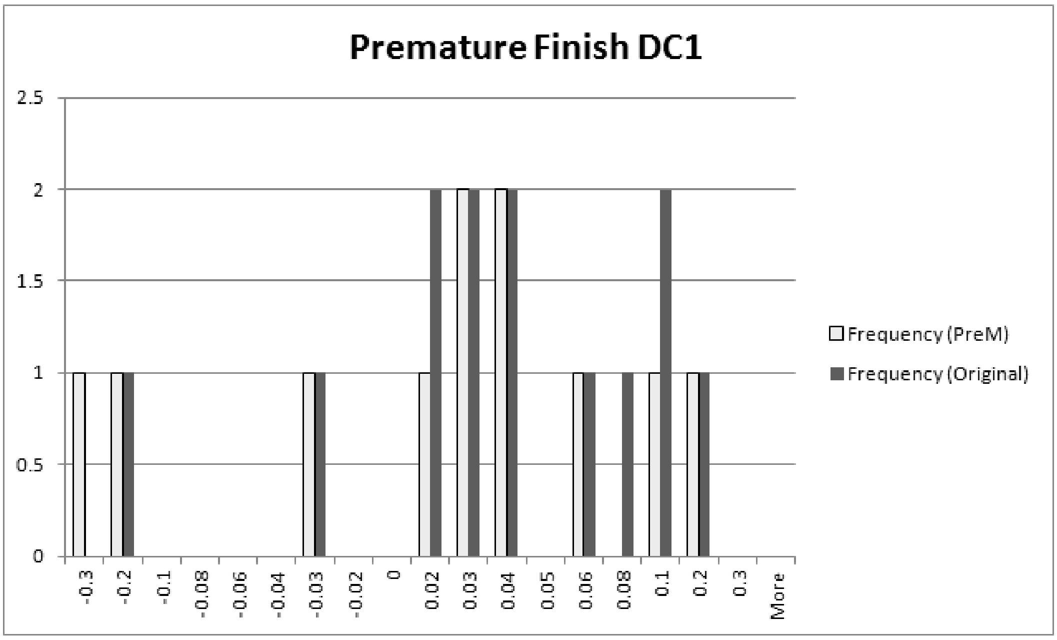

Figure 27: Comparison of CSV with premature finish addition to original CSV algorithm using DC1 data.

CSV Premature Finish DC3 (Field)

The premature finish addition to the CSV application was implemented because of the problem in two of the events in this data set. The first event was overestimated by 19.3% by the original version of the algorithm. The premature addition estimated the weight only 2.6% over the true weight. The result for this event now falls within the desired accuracy for the algorithm.

The CSV application with the premature finish addition estimated 27 weights, compared to the 29 with the original version. It improved two of the results, but made three worse and missed two.

The premature finish and the original version of the application had the same estimated weights for 22 of the events. This was expected, as the premature finish addition should only make a difference to events which dropped below the threshold early. The premature finish version can run into problems when animals follow each other too closely. It can have trouble distinguishing events if the weight is below the threshold for less than 240ms, between the end of the first and the start of the second animal.

Table 11: Comparison of CSV premature finish to original CSV algorithm using DC3 data.

Bin	*Frequency (PreM)*		*Frequency (Original)*	
-0.3	0	0%	0	0%
-0.2	0	0%	0	0%
-0.1	0	0%	0	0%
-0.08	0	0%	1	3%
-0.06	1	4%	0	0%
-0.04	2	7%	2	7%
-0.03	0	0%	0	0%
-0.02	0	0%	0	0%
0	1	4%	2	7%
0.01	1	4%	1	3%
0.02	2	7%	2	7%
0.03	4	15%	4	14%
0.04	7	26%	7	24%
0.05	1	4%	1	3%
0.06	2	7%	2	7%
0.07	3	11%	3	10%
0.08	0	0%	0	0%
0.1	0	0%	0	0%
0.2	2	7%	3	10%
0.3	0	0%	1	3%
More	1	4%	0	0%
Total	27	100%	29	100%
Within 3%	8	30%	9	31%

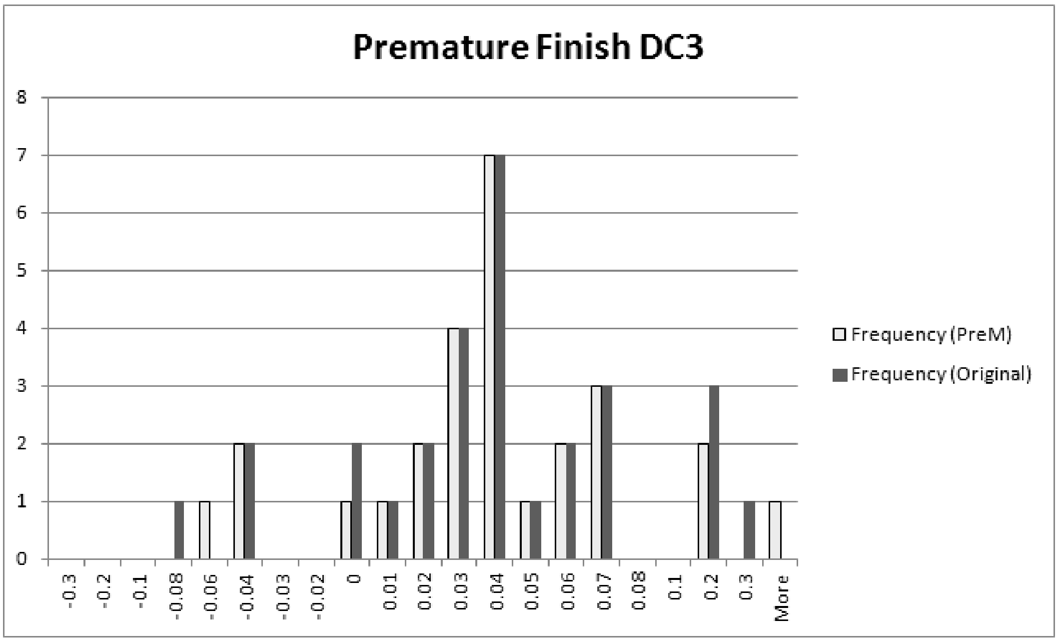

Figure 28: Comparison of CSV with premature finish addition to original CSV algorithm using DC3 data.

4.4.3 Conclusion

It was found that the CSV application with the pre-smoothing addition had a tendency to underestimate most of the weights. The premature finish addition was able to help in a few cases, but did drop the overall accuracy marginally.

Pre-smoothing Addition

The pre-smoothing was ineffective as it lowered the accuracy and processed fewer events. The smoothing caused most of the results to be estimated at lower weights and below their true weight. This resulted in the outcome being worse than the original CSV application estimates. The accuracy (within 3%) dropped by 13% and 17% in these example above.

Premature Finish Addition

The premature finish addition to the CSV application was able to improve one of the events, for which the addition was made for. Unfortunately it did have some negative effects on some of the events that were not an issue. The overall the accuracy (within 3% of true weight) dropped by 1-2% compared to the original CSV application.

4.5 Code for DairyScale Hardware

The initially there were two ideas for the code that would be written for the DairyScale hardware. The first idea was to make adjustments to the DairyScale code. The second idea was to write an algorithm specifically designed for dry stock.

The code on the DairyScale was designed to weigh dairy cows, so would need adjustments made so it would be suitable for dry stock cattle. The main change would be to alter the error checking so it is more suited to the weighing of dry stock. For example one of the error checks would need to be removed, as it was based around seeing animals twice a day. This particular error check only works after the animal has walked over the WOW 12 times.

During the course of this research an algorithm was devised specifically for dry stock cattle, from the data collected from cattle walking over the WOW. The algorithm was implemented into an application, which allows it to process a CSV file with raw data from the WOW system.

This algorithm implemented in the application has not been implemented on the DairyScale hardware yet.

5.0 Conclusion and Recommendations

This research investigated the possibility of creating a system that could allow unsupervised weighing of dry stock cattle. Based on an existing product, Gallagher's DairyScale, the option of using this unmodified was explored and was compared to an algorithm which was devised for dry stock cattle. An application was created to trial the new dry stock algorithm on the data collected from the field trials, enabling us to see the effectiveness of the new algorithm and compare it against the existing algorithm.

In conclusion to this work a summary of the limitations is given and recommendations for future work are suggested.

5.1 Accomplishments

The first achievement was collecting raw data of dry stock cattle walking over the walk-over weighing platform. These data collections also include DairyScale weight estimation results, static weights, and some video footage. These data collections include valuable data for this research and any future work that might be carried out.

It was found the DairyScale's accuracy was nearing an acceptable range for some of the herds, although it did disregard a large proportion of the animals. The behaviour of the animals caused them to not have weights estimated, particularly the animals following each other too closely.

The new algorithm estimated weights for a greater proportion of the cattle compared to the DairyScale algorithm, although the weight estimations were not as accurate.

The success of the new algorithm was that it did not require the use of the lead-on platform for locating when an animal was stepping onto the WOW. The EID tag read was used to locate when an animal was coming on to the main platform.

5.2 Algorithm

The new algorithm for dry stock cattle has several advantages and disadvantages, over the DairyScale algorithm. The behaviour of the dry stock cattle caused the data to be unclear, which made it difficult to work with.

DairyScale algorithm used the lead on platform to detect an animal coming onto the WOW. The new algorithm does not require the lead-on. The new algorithm uses the EID tag to identify an animal coming onto the WOW and a rise above the threshold as the start of an event.

Dry stock cattle have two traits which often cause the data to be unclear. The speed they travel at and the closeness they do it at when they travel in groups. This is possibly caused by being uncomfortable around the new equipment.

The new algorithm produces weight estimations for a greater proportion of events, than the DairyScale algorithm does. The problem with some of the data received is its quality, making it very difficult to estimate the weight accurately. This causes estimations to fall outside of the 3% accuracy which is required. The error checking of the DairyScale reduces the number of inaccurate weight estimations, by dropping the events with low quality data.

Two additions were made to the original sum threshold algorithm, in an attempt to fix a problem and improve the accuracy of the weight estimations.

The first was to run a moving window average over the data to smooth the data before applying the algorithm on the events. Pre-smoothing had very negative effect on the accuracy of the results, underestimating the majority of the weights.

The second addition to the code was to correct a problem when an event was cut short by the algorithm. The situation occurred only a few times and was caused by the weight dropping below the threshold for a fraction of a second. The addition worked as intended on the majority of the events, but did have an unintended negative effect on some results. Overall the accuracy was slightly lowered by 1 - 2%.

The three different versions of the new algorithm did not achieve the required accuracy, and would require further development to be suitable for implementation for the end users.

5.3 Limitations

During the research several limitations and problems arose that had to be overcome to complete the research. The main issues were the weather conditions, equipment failure and animal behaviour.

The weather condition had a large impact on how often data collection sessions could be performed (see 3.3.4 Field Data Collection). There had to be a long enough break in the rain so cattle would not cause too much damage to the ground when walking over it numerous times. The days that were too wet were used to analyse previously collected data and develop the algorithm.

It would have been preferable to collect more field data, unfortunately equipment failure during the field data collection infringed on the possibility of getting any additional complete data sets (see 3.3.5 Field Data Collection – Wider Range / Lighter Animals). The SmartReader did have some problems reading EID tags; the effect of this was that events in the data could not be matched to the animals' static weight. The team at Gallagher's was able to test and troubleshoot the hardware, to find the source of the equipment faults.

When running the equipment the batteries and solar panels were not able to supply enough power to keep the system running for long periods of time (see 3.1.1 WOW Crate). Resulting in data collections being cut short, therefore data collection sessions could not be taken over several days. It was discovered that it was better to carry out short data collections, as they provided visual data that was invaluable.

Several cattle had two different EID tags on them, which was not expected. To work around this when processing the data with the CSV application, a modification was made so these invalid tags were ignored (see 4.2.1 Skipping Invalid EID Tags). The code in the CSV application would not be in the final product, as some farmers would use tags in that range.

The erratic behaviour of the animals was an issue, as they did not want to cross the WOW and when they did cross they often did it in closely spaced groups (see 2.3.1 Challenges of Dry Stock). This resulted in the animals being too close to each other, causing some of the data to be difficult to use or not usable at all.

The new cattle algorithm was not implemented on the DairyScale hardware. Due to the amount of time that was consumed collecting data. It would have been preferable to test the new algorithm on the DairyScale hardware in the field. The alternative method of testing the performance of the new algorithm was to use the data collected throughout the project.

5.4 Future Work

During the research it became clear that improvements could be made at several stages of the weighing process, which could be investigated further. The following points describe the possible further improvements and future work.

1. Flexible threshold
    a. An adaptable threshold could be developed so various weight ranges of cattle can be weighted with the WOW. This could either be a setting that is selected by the farmer, or the algorithm could decide on the threshold to use.
2. Rising and falling edge
    a. Develop an algorithm which locates a rising and falling edge, rather than using a threshold for locating the start and end points of each walkover event. This system could be better at recognising when an animal is fully on the platform, but could be more computationally intensive. This could also discard events where the animals are crossing the platform too close together.
3. Develop a filter
    a. Investigate if a filter for discarding or marking events which have a high chance of being inaccurate. This filter would remove or mark erroneous events that are caused by animals following too closely, erratic behaviour or fast movement over the weighing platform.
4. Animal behaviour

    a. Research the use of animal separators or chicanes to improve the animals' behaviour on the WOW. The expectation is that these tools could improve the quality of the data therefore making the weight estimations more accurate.
    b. It could be worth researching an option of using ramps in front and behind the weighing platform to give smoother transitions and clearer data.
5. DairyScale
    a. Investigate if a less intensive error checking system would improve the number of valid weight estimations produced by the DairyScale algorithm for dry stock cattle.

5.5 Conclusion

The research investigated the possibility of using the existing DairyScale equipment to weight dry stock cattle, by either modifying the existing DairyScale code or creating a new algorithm that would be implemented on the DairyScale hardware.

The farmers and research organisations requested an automatic walk-over weight system for dry stock, similar to the one used for dairy cows. This product would give farmers a more efficient method of weighing and be less stressful for the cattle.

The DairyScale algorithm and the new algorithm specifically developed for dry stock cattle were both trialled on multiple herds of cattle.

Neither algorithm was able to achieve the desired weighing accuracy, but the new algorithm did estimate weights in the majority of cases. Additionally the new algorithm was successful in eliminating the need for the lead-on platform for detecting animals.

The input data quality could be improved with better management of the dry stock cattle while they walk-over the weighing equipment. This could be achieved by a better understanding of the cattle behaviour, and the use of a chicane or animal separator located before the WOW. These improvements and further tuning of the algorithm could make the use of the DairyScale hardware suitable for weighing dry stock cattle.

References

Alawneh, JI. (2011). *Monitoring liveweight to optimise health and productivity in pasture fed dairy herd.* (Doctoral dissertation, Massey University, Palmerston North, New Zealand). Retrieved from http://mro.massey.ac.nz/bitstream/handle/10179/3167/01_front.pdf?sequence=1

Chesterman, J. (1842). *Chesterman's Cattle Gauge*. Retrieved January 10, 2013, from Nathan Zelders: http://www.nzeldes.com/HOC/CattleGauge.htm

Gallagher Group Limited. (2011, April). *DairyScale Installer Manual.* Retrieved April 18, 2012, from Gallagher Animal Management: http://www.gallagher.co.nz/weighing-and-eid-manuals.aspx

Gallagher Group Limited. (2011, September). *DairyScale XDS5000 User manual.* Retrieved April 13, 2012, from Gallagher Animal Management: http://www.gallagher.co.nz/weighing-and-eid-manuals.aspx

Gallagher Group Limited. (n.d.). WOW high level design.

GEA Farm Technologies. (n.d.). *Taxatron 5000*. Retrieved April 15, 2013, from GEA Farm Technologies: http://www.westfalia.com/au//en/bu/milking_cooling/farm_management/scales/taxatron_5000/default.aspx

Harris, M. (2011). *Automatic Weighing for Drystock: CONFIDENTIAL.* Hamilton: Gallagher Group Limited.

Hyperdrug Pharmaceuticals Ltd. (n.d.). *Weight Measuring Tape For Cattle & Pigs*. Retrieved March 27, 2013, from Hyperdrug: https://www.hyperdrug.co.uk/Weight-Measuring-Tape-For-Cattle-Pigs/productinfo/WEIGHTTAPE/

King, B. (2012, April 6). *meat: growing a milk cow (Family milk cow project).* Retrieved April 23, 2013, from meat: http://ebeyfarm.blogspot.co.nz/2012/04/growing-milk-cow-family-milk-cow.html

NAIT Ltd. (n.d.). *Tagging requirements*. Retrieved February 27, 2013, from National Animal Identification and Tracking Scheme:

http://www.nait.co.nz/news-and-publications/nait-user-guides-and-fact-sheets/tagging-requirements/

Pater, S. (2007). How Much Does Your Animal Weigh? *Backyards & Beyond , 1* (3), p. 11-12.

Teal, P. *Dynamic Algorith Design.*

Tru-Test Group. (n.d.). *Indicators >WOW! XR3000- Tru-Test livestock weigh scales animal weighing equipment*. Retrieved January 3, 2013, from Tru-Test Group - Made to Measure: http://weighing.tru-test.com/weigh_scale_details.asp?&product=WOW!%20XR3000&pid=2282

Appendices

Appendix A – DVD

Folders:

- User manuals
  - DairyScale - User manual.
  - DairyScale - Installer manual.
- Application
  - Source file.
  - Source code (PDF).
- Collected Data
  - In respective folders, see Table 12: Data collections.

Table 12: Data collections.

Data collection	Date	Herd	Location	No. of cattle	Runs	Static weight-File name
DC1	11/06/2012	Herd1	Stockyard	26	5	
DC1	11/06/2012	Herd2	Stockyard	41	2	
DC1 (Day 2)	12/06/2012	Herd3	Stockyard	25	3	
DC1 (Day 2)	12/06/2012	Herd4	Stockyard	45	2	
DC2	1/08/2012		Stockyard	30	3	
DC3	1/08/2012		Field	30	1	
DC4	2/08/2012		Field	30	1	7 EID's recorded by DairyScale
DC5	21/11/2012		Field	46	1	No EID's recorded by PC
DC6	1/03/2013		Field	NA	Part	2 DairyScale events with EID
DC7	4/03/2013		Stockyard	52	1	Short range reading of EID tags
DC8	11/03/2013		Stockyard	52	2	Error in EID's recorded by PC
DC9	21/03/2013		Stockyard	NA	3	No EID's recorded by PC

Appendix B – Project Proposal

Dry Stock Walkover Weighing - Research Project Proposal

Gallagher Background

Gallagher manufacture animal management systems, fuel systems, and security products.

Project Background

Most animal weighing systems require the animal to be restrained while the weight reading is being taken.

In conjunction with Gallagher, Paul Teal devised an algorithm which can estimate the weight of an animal while it is walking over a set of scales (load bars). The product is called DairyScale. It is comprised of several parts, including load bars, electronic identification (EID) sensor, and a control box for calculating and recording the data. The EID sensor is able to identify a particular dairy cow, so the weight can linked to the cows ID and be tracked over time.

Gallagher's DairyScale product is designed to weigh moving dairy cows. Dairy cows are easy to weight, as they are placid animals. Measuring the weight of other moving animals is more difficult. This projects aim is to investigate the possibility of weighing moving dry stock. This is to be done using the existing DairyScale hardware, and adjust the existing algorithm to suit dry stock.

Objectives

Investigate the possibility of adapting Gallagher's existing DairyScale to weigh dry stock.

Aims:

- Main: Ability for the existing device to measure weights of moving dry stock (such as cattle or calves)

- Bonus: Remove one of the three weight sensors
    - To make a cheaper version of the product (Can be limited to only dry stock)

Tasks:

- Become familiar with equipment (performance, limitations)
    - Use equipment in the field
- Become familiar with the existing algorithm
    - Talk to people who are knowledgeable with the code
- Gather weight data (Raw data and actual weights)
- Analyse weight data
    - Explore possibility of using WEKA or Adams to identify an algorithm for dry stock
- Adjust existing code to suit new algorithm and implement on the device
    - Review performance of the equipment and its suitability for the new algorithm
    - Test new algorithm in lab
    - Evaluate suitability of algorithm
- Test new code in field
    - Confirm performance and limitations (in the field)

Methodology

Stage 1

- Project proposal
- Become familiar with existing device, algorithm and development environment
- Talk to people familiar with existing equipment and software
- Get familiar with existing raw data

Stage 2

- Decide on hardware to gather data
- Consult with Paul Teal (about hardware and setup of testing trailer)
- Run test of equipment
- Consult with Paul Teal (results useable)
- Field work – gather raw weight data

Stage 3

- Analyse data
- Write test program
- Test in lab

Stage 4

- Implement on device
- Test in lab
- Test in the field
- Confirm results accuracy

Stage 5

- Implement on device
- Test in lab
- Test in the field
- Finalise code
- Final testing

Stage 6

- Write final report
- Get final report checked by my supervisor
- Make changes to report (several iterations)
- Submit final report

There will be iterations on several of the stages above.

Timeline

March – Project proposal.

April – Become familiar with current device, algorithm and development environment.

May – Field work – Gathering weight information.

June – Analyse weight data.

July – Analyse weight data.

August – Implement new algorithm on current device.

September – Implement new algorithm on current device.

October – Implement new algorithm on current device.

November – Finalise and test code on hardware.

December – Write thesis.

January – Write thesis.

February – Write thesis.

March – Write thesis.

Appendix C – Weight Estimation Results

Data A

Electronic Tag	Weight (kg)	Est W1	Est W2	Est W3	Diff W1	Diff W2	Diff W3	% W1	% W2	% W3
942000003525450	465	0			465			100%		
942000003525387	477	484			-7			-1%		
982000146500578	481	487	0		-6	481		-1%	100%	
942000003525480	500	480	0		20	500		4%	100%	
982000146500589	468									
942000003525401	514	523	0		-9	514		-2%	100%	
982000146500330	488									
942000003525388	456	477	269		-21	187		-5%	41%	
942000003525397	483	476	460		7	23		1%	5%	
982000146500602	409	452			-43			-11%		
942000003525403	462	467	468		-5	-6		-1%	-1%	
942000003525468	476	0	489		476	-13		100%	-3%	
942000003525696	477	488	488		-11	-11		-2%	-2%	
942000003525627	483	443	473		40	10		8%	2%	
942000003525469	484	0	491		484	-7		100%	-1%	
982000146500157	530	0	484		530	46		100%	9%	
942000003525422	502	0	433		502	69		100%	14%	
982000146500140	467	0	414		467	53		100%	11%	
942000003525451	429	251	427		178	2		41%	0%	
942000003525394	510	0	536		510	-26		100%	-5%	
942000003525481	457	467	463		-10	-6		-2%	-1%	
942000003525410	474	482	476		-8	-2		-2%	0%	
942000003525428	479	480	441		-1	38		0%	8%	
942000003525396	492	0	306		492	186		100%	38%	
982000146500581	483									
982000146500646	475	0	0		475	475		100%	100%	
942000003525609	478	485	491		-7	-13		-1%	-3%	
942000003525416	444	0	0		444	444		100%	100%	
982000146513575	450	469	501		-19	-51		-4%	-11%	
942000003525688	465	487	0		-22	465		-5%	100%	
982000146500677	467	465	490		2	-23		0.4%	-5%	
942000003525419	466	491	467		-25	-1		-5%	0%	
942000003525462	499	0	501		499	-2		100%	0%	
982000146500197	461	0			461			100%		
951000010629211										
942000003525698	470	451	468		19	2		4%	0%	
942000003525471	467	0	0		467	467		100%	100%	
982000146500604	496	0			496			100%		
942000003525424	470	474			-4			-1%		
982000146500621	439	0			439			100%		
942000003525405	485	485	294	469	0	191	16	0%	39%	3%
942000001754744	392	396			-4			-1%		
942000003525482	445	423	0		22	445		5%	100%	
982000146500698	483	0	485		483	-2		100%	0%	
942000003525457	504	502			2			0%		
942000003525406	442	448	0		-6	442		-1%	100%	

Data B

	Static	DairyScale Estimation			Difference (Kgs)			Percentage Difference		
Electronic Tag	Weight (kg)	Est 1	Est 2	Est 3	Diff 1	Diff 2	Diff 3	% Diff 1	% Diff 2	% Diff 3
942 000003525450	465	X								
942 000003525387	477	484			-7			-1%		
982 000146500578	481	487			-6			-1%		
942 000003525480	500	480			20			4%		
982 000146500589	468									
942 000003525401	514	523			-9			-2%		
982 000146500330	488									
942 000003525388	456	477	269		-21	187		-5%	41%	
942 000003525397	483	476	460		7	23		1%	5%	
982 000146500602	409	452			-43			-11%		
942 000003525403	462	467	468		-5	-6		-1%	-1%	
942 000003525468	476		489			-13			-3%	
942 000003525696	477	488	488		-11	-11		-2%	-2%	
942 000003525627	483	443	473		40	10		8%	2%	
942 000003525469	484		491			-7			-1%	
982 000146500157	530		484			46			9%	
942 000003525422	502		433			69			14%	
982 000146500140	467		414			53			11%	
942 000003525451	429	251	427		178	2		41%	0%	
942 000003525394	510		536			-26			-5%	
942 000003525481	457	467	463		-10	-6		-2%	-1%	
942 000003525410	474	482	476		-8	-2		-2%	0%	
942 000003525428	479	480	441		-1	38		0%	8%	
942 000003525396	492		306			186			38%	
982 000146500581	483									
982 000146500646	475									
942 000003525609	478	485	491		-7	-13		-1%	-3%	
942 000003525416	444									
982 000146513575	450	469	501		-19	-51		-4%	-11%	
942 000003525688	465	487			-22			-5%		
982 000146500677	467	465	490		2	-23		0%	-5%	
942 000003525419	466	491	467		-25	-1		-5%	0%	
942 000003525462	499		501			-2			0%	
982 000146500197	461									
951 000010629211										
942 000003525698	470	451	468		19	2		4%	0%	
942 000003525471	467									
982 000146500604	496									
942 000003525424	470	474			-4			-1%		
982 000146500621	439									
942 000003525405	485	485	294	469	0	191	16	0%	39%	3%
942 000001754744	392	396			-4			-1%		
942 000003525482	445	423			22			5%		
982 000146500698	483		485			-2			0%	
942 000003525457	504	502			2			0%		
942 000003525406	442	448			-6			-1%		

Data C

		CSV Application Estimations				Difference (Kgs)				Percentage Difference			
Electronic Tag	Weight (kg)	Est W1	Est W2	Est W3	Est W4	Diff W1	Diff W2	Diff W3	Diff W4	% W1	% W2	% W3	% W4
942000003055858	366	0	362	371		366	4	-5			1%	-1%	
942000003055833	354	0	0	0		354	354	354			100%	100%	
942000003035664	334	0	0			334	334				100%		
942000003035577	361	363	0			-2	361			-1%	100%		
942000003030127	351	390	0	377		-39	351	-26		-11%	100%	-7%	
942000003035550	356	0	0	357	356	356	356	-1	0		100%	0%	0%
942000003028038	360	0	356	0	358	360	4	360	2		1%	100%	1%
942000003035567	348	0				348							
942000003030103	380	376	353	362	0	4	27	18	380	1%	7%	5%	
942000003035554	358	0				358							
942000003035679	369	0	0	372		369	369	-3			100%	-1%	
942000003042953	354	0	353			354	1				0%		
942000003525700	360	360	303	361		0	57	-1		0%	16%	0%	
942000003042984	362	0	0	361		362	362	1			100%	0%	
942000003028001	372	358	0	0		14	372	372		4%	100%	100%	
942000003030117	378	380	0	0		-2	378	378		-1%	100%	100%	
942000003042966	366	0				366							
942000003042985	360	356	0	0	462	4	360	360	-102	1%	100%	100%	-28%
942000003055611	353	0	395			353	-42				-12%		
942000003041074	356	362				-6				-2%			
942000003035561	373	0	376			373	-3				-1%		
942000003042952	346	0	0	348	351	346	346	-2	-5		100%	-1%	-1%
942000003035560	357	0	0	0	368	357	357	357	-11		100%	100%	-3%
942000003055635	351	0	0	0		351	351	351			100%	100%	
942000003035572	348	0				348							
942000003041012	322	0				322							
942000003028036	353	0				353							
942000003055877	362	368				-6				-2%			
942000003055688	346	0				346				100%			
942000003035589	393	0	0			393	393			100%	100%		

Data D

		DairyScale Estimates			Difference			Percentage Difference		
Electronic Tag	Weight (kg)	Est 1	Est 2	Est 3	Diff W1	Diff W2	Diff W3	% W1	% W2	% W3
942000003055858	366	479.9	377.6	379.2	-113.885	-11.635	-13.1621	-31%	-3%	-4%
942000003055833	354	363.5	371.7	370.1	-9.51	-17.6643	-16.1201	-3%	-5%	-5%
942000003035664	334	511.0	423.0	342.2	-176.985	-89.035	-8.151	-53%	-27%	-2%
942000003035577	361	395.5	388.0		-34.5172	-26.969		-10%	-7%	
942000003030127	351	401.1	384.4	375.2	-50.06	-33.4488	-24.1557	-14%	-10%	-7%
942000003035550	356	457.0	372.8		-100.985	-16.7906		-28%	-5%	
942000003028038	360	386.2	209.6	410.3	-26.2241	150.3608	-50.2597	-7%	42%	-14%
942000003035567	348	368.2	388.6	323.7	-20.2144	-40.5725	24.2612	-6%	-12%	7%
942000003030103	380	401.8	393.6		-21.8183	-13.6223		-6%	-4%	
942000003035554	358	438.5	387.1	371.1	-80.5123	-29.0975	-13.0695	-22%	-8%	-4%
942000003035679	369	397.6	400.2	404.6	-28.5735	-31.1821	-35.625	-8%	-8%	-10%
942000003042953	354	365.6	367.6	498.0	-11.557	-13.62	-144.004	-3%	-4%	-41%
942000003525700	360	382.9	376.1		-22.9077	-16.1207		-6%	-4%	
942000003042984	362	378.9	619.8	431.7	-16.9428	-257.835	-69.6996	-5%	-71%	-19%
942000003028001	372	467.6	420.6		-95.585	-48.5753		-26%	-13%	
942000003030117	378	378.7	388.4	392.7	-0.7386	-10.3697	-14.7402	0%	-3%	-4%
942000003042966	366	536.6			-170.585			-47%		
942000003042985	360	395.5	378.1	377.0	-35.495	-18.1017	-16.9647	-10%	-5%	-5%
942000003055611	353	383.9			-30.915			-9%		
942000003041074	356	370.8	410.5	351.0	-14.7591	-54.535	4.965	-4%	-15%	1%
942000003035561	373	388.9	594.8		-15.9038	-221.793		-4%	-59%	
942000003042952	346	414.0	413.1	359.9	-67.965	-67.1025	-13.9292	-20%	-19%	-4%
942000003035560	357	390.9	391.5		-33.9134	-34.5245		-9%	-10%	
942000003055635	351	471.6	501.2	369.6	-120.585	-150.175	-18.6091	-34%	-43%	-5%
942000003035572	348	358.6	401.0	464.1	-10.5893	-52.985	-116.092	-3%	-15%	-33%
942000003041012	322	343.9	340.7		-21.9109	-18.7228		-7%	-6%	
942000003028036	353	476.1	410.7	410.3	-123.135	-57.735	-57.285	-35%	-16%	-16%
942000003055877	362	394.3	380.7	380.9	-32.3322	-18.7066	-18.8753	-9%	-5%	-5%
942000003055688	346	372.3	363.9	358.1	-26.315	-17.9225	-12.0979	-8%	-5%	-3%
942000003035589	393	324.3	413.4	409.7	68.7287	-20.4067	-16.6988	17%	-5%	-4%

Data E

Electronic Tag	Weight (Kg)	Notes	Est (Kg)	Diff	%
942000003055877	362	942000003055877	367	5	1%
951000010784060		rouge	361		
942000003042984					
942000003055833	354	942000003055833	376	22	6%
951000009588632					
951000015845676		rouge	368		
942000003035560					
942000003030103	380	942000003030103	384	4	1%
951000014908296					
942000003042985	360	942000003042985	361	1	0%
942000003055635	351	942000003055635	287	-64	-22%
942000003035664					
951000017045073					
942000003035550					
942000003028001	372	942000003028001	272	-100	-37%
942000003042953					
942000003041074	356	942000003041074	335	-21	-6%
951000015554586					
951000014395582					
951000014760158		rouge	346		
942000003035577	361	942000003035577	365	4	1%
951000013280865					
942000003041012	322	942000003041012	316	-6	-2%
942000003055611					
942000003030127	351	942000003030127	349	-2	-1%
942000003035572	348	942000003035572	341	-7	-2%
951000014982400					
942000003042966	366	942000003042966	351	-15	-4%
942000003035679	369	942000003035679	378	9	2%

Data F

Electronic Tag	Weight (kg)	Estimate	Difference	Error
3055877	362	378.4	16.4	5%
10784060				
3042984	362	374	12	3%
9394104				
3055833	354	367.8	13.8	4%
3055858	366	377.3	11.3	3%
3035561	373	381.1	8.1	2%
9588632				
3042952	346	343.1	-2.9	-1%
15845676				
3525700	360	364.3	4.3	1%
672975				
3055688	346			
9957152				
3028036	353	335.6	-17.4	-5%
3035560	357	455.4	98.4	28%
3030103	380	387.4	7.4	2%
3030117	378	400.9	22.9	6%
14908296				
3042985	360	372.8	12.8	4%
3055635	351	348.5	-2.5	-1%
3035664	334	343.1	9.1	3%
3035589	393	417.1	24.1	6%
17045073				
3035554	358	367.8	9.8	3%
3035550	356	368.3	12.3	3%
3028001	372	393.6	21.6	6%
16143529				
3042953	354	364.1	10.1	3%
3041074	356	408.3	52.3	15%
15554586				
3035567	348	329.8	-18.2	-5%
14760158				
3028038	360	371.3	11.3	3%
14395582				
3035577	361	383.3	22.3	6%
13280865				
3041012	322	384.2	62.2	19%
3055611	353	321.9	-31.1	-9%
3030127	351	364.4	13.4	4%
3035572	348	350.6	2.6	1%
14982400				
3042966	366	433.8	67.8	19%
3035679	369	390.6	21.6	6%
16159554				

Data DC1 – CSV with Additions DC1

	Static	CSV App			Smoothing short				Premature		
Electronic Tag	Weight (k	Estimate	Diff	%	Est	Diff	%	Notes	Est	Diff	%
942 000003525450	465	495.5	30.5	7%				miss			
942 000003525387	477										
982 000146500578	481										
942 000003525480	500										
982 000146500589	468										
942 000003525401	514										
982 000146500330	488										
942 000003525388	456	345.1	-110.9	-24%				miss	345.1348	-110.865	-24%
942 000003525397	483										
982 000146500602	409										
942 000003525403	462										
942 000003525468	476										
942 000003525696	477										
942 000003525627	483	526.5	43.5	9%				miss	325.6585	-157.342	-33%
942 000003525469	484										
982 000146500157	530										
942 000003525422	502										
982 000146500140	467										
942 000003525451	429	440.7	11.7	3%	335.795	-93.205	-22%	Bad	440.7067	11.7067	3%
942 000003525394	510										
942 000003525481	457	495.1	38.1	8%	464.7577	7.7577	2%	Closer	495.0785	38.0785	8%
942 000003525410	474										
942 000003525428	479										
942 000003525396	492										
982 000146500581	483										
982 000146500646	475										
942 000003525609	478	460.8	-17.2	-4%				miss	460.8268	-17.1732	-4%
942 000003525416	444	451.7	7.7	2%	199.8016	-244.198	-55%	Bad	451.7223	7.7223	2%
982 000146513575	450										
942 000003525688	465	482.2	17.2	4%				miss	482.1656	17.1656	4%
982 000146500677	467										
942 000003525419	466	489.8	23.8	5%	485.5125	19.5125	4%	closer	489.8081	23.8081	5%
942 000003525462	499										
982 000146500197	461										
951 000010629211											
942 000003525698	470	520.5	50.5	11%	361.4675	-108.533	-23%	Bad	520.4908	50.4908	11%
942 000003525471	467	480.3	13.3	3%	359.8691	-107.131	-23%	Bad	480.3331	13.3331	3%
982 000146500604	496										
942 000003525424	470	488.4	18.4	4%	391.2007	-78.7993	-17%	Bad	488.3951	18.3951	4%
982 000146500621	439										
942 000003525405	485										
942 000001754744	392										
942 000003525482	445										
982 000146500698	483										
942 000003525457	504	504.9119	0.9119	0.001809	493.8009	-10.1991	-2%	Good			
942 000003525406	442										

Data DC3 – CSV with Additions DC3

Tag	Static	CSV		error		Premature	PreM to CSV	PreM to true		Smoothing			
	Weight (kg)	Estimate	Diff	%	Notes	Estimate	Diff	Diff	Notes	Estimat	Diff	% Diff	Notes
3055877	362	378.4	16.4	5%		378.4	0.0	4.5%		286.6	-75.4	-21%	
10784060													
3042984	362	374.0	12.0	3%		374.0	0.0	3.3%					
9394104													
3055833	354	367.8	13.8	4%		367.8	0.0	3.9%		304.4	-49.6	-14%	
3055858	366	377.3	11.3	3%		377.3	0.0	3.1%		218.0	-148.0	-40%	
3035561	373	381.1	8.1	2%		381.1	0.0	2.2%		361.8	-11.2	-3%	
9588632										314.1			
3042952	346	343.1	-2.9	-1%		319.0	-24.1	-7.8%	Bad				
15845676													
3525700	360	364.3	4.3	1%		364.3	0.0	1.2%					
672975													
3055688	346			-100%	Tag too late								
9957152													
3028036	353	335.6	-17.4	-5%		414.9	79.3	17.5%	Bad	406.1	53.1	15%	
3035560	357	455.4	98.4	28%	Mix up with animals								
3030103	380	387.4	7.4	2%		387.4	0.0	1.9%					
3030117	378	400.9	22.9	6%		400.9	0.0	6.1%					
14908296													
3042985	360	372.8	12.8	4%		372.8	0.0	3.6%					
3055635	351	348.5	-2.5	-1%		348.5	0.0	-0.7%					
3035664	334	343.1	9.1	3%		470.7	127.6	40.9%	Bad	291.5	-42.5	-13%	
3035589	393	417.1	24.1	6%		417.1	0.0	6.1%					
17045073													
3035554	358	367.8	9.8	3%		367.8	0.0	2.7%		306.0	-52.0	-15%	
3035550	356	368.3	12.3	3%		368.3	0.0	3.5%		247.9	-108.1	-30%	
3028001	372	393.6	21.6	6%		393.6	0.0	5.8%		304.1	-67.9	-18%	
16143529													
3042953	354	364.1	10.1	3%		364.1	0.0	2.8%		329.6	-24.4	-7%	
3041074	356	408.3	52.3	15%		408.3	0.0	14.7%		355.3	-0.7	0%	Good
15554586													
3035567	348	329.8	-18.2	-5%		329.8	0.0	-5.2%					
14760158													
3028038	360	371.3	11.3	3%		371.3	0.0	3.1%					
14395582													
3035577	361	383.3	22.3	6%		383.3	0.0	6.2%					
13280865													
3041012	322	384.2	62.2	19%	Dropped	330.4	-53.9	2.6%	Good	285.9	-36.1	-11%	Improved
3055611	353	321.9	-31.1	-9%		336.4	14.5	-4.7%	Better				
3030127	351	364.4	13.4	4%		364.4	0.0	3.8%		294.4	-56.6	-16%	
3035572	348	350.6	2.6	1%		350.6	0.0	0.7%		314.3	-33.7	-10%	
14982400													
3042966	366	433.8	67.8	19%	Dropped below threshold				MISSED	283.3	-82.7	-23%	
3035679	369	390.6	21.6	6%		390.6	0.0	5.9%					

Appendix D – Code of CSV Application

```
Project: CSV Reader_WeigherV2.rbp
Date: Tuesday, 15 January 2013 1:58:42 PM
Project Info:
Mac (Carbon PEF) App Name: Read_CSV
Mac (Carbon Mach-O) App Name: My Application
Mac (Classic) App Name: My Application (Classic)
Windows App Name: My Application.exe
Linux App Name: MyApplication
Long Version:
Major Version: 1
Minor Version: 0
Sub Version: 0
Release: 0
Non-Release: 0
Mac Creator Code:
Windows MDI Caption:
Minimum Memory Size: 2048
Standard Memory Size: 4096
Class Read_CSV
Inherits Application
Const kFileQuitShortcut = ""
Const kFileQuit = "&Quit"
Const kEditClear = "&Delete"
End Class
Class FileReader
Inherits Window
FileReader.Table(5,150000):
Table(5,150000) As Double
DataLines As Integer
Offset1 As Double
Offset2 As Double
Threshold As Double
FilePosition As Integer
EarTag As Integer
EarTagFlag As Boolean
Untitled As Integer
Line As Integer
Column As Integer
Untitled1 As Integer
TagMax As Double
TagMin As Double
Untitled2 As Integer
WeighLine As Integer
Untitled3 As Integer
EarTagPosn As Integer
Untitled4 As Integer
FileReader Control OpenFile:
Sub Action()
'FileReader properties
'Offset1, Offset2 As Double - Load bar zero offsets calculated here
'Threshold As Double - calculate threshold for animal presence here
'FilePosition As Integer - current line in file for starting next scan
'Table(5,150000) As Double - array holding 1 hours WOW data
'DataLines As Integer - number of valid data lines in Table
'EarTag As Integer - last read ear tag value which may have to transfer from one
cow
to the next
'EarTagFlag As Boolean - an indicator that there is a current ear tag
Dim f as FolderItem
Dim tis as TextInputStream
Dim s As String
Dim i As Integer
Dim fields() as String
'show standard file selector
```

```
f = GetOpenFolderItem("any")
if f=nil then exit sub 'cancel clicked
'open the file
tis = f.OpenAsTextFile
if tis = nil then 'open failed
MsgBox("The file could not be opened")
exit sub
end if
'read file into grid
while not tis.EOF
DisplayGrid.AddRow""
s = tis.ReadLine
fields = Split(s,",")
for i = 0 to DisplayGrid.ColumnCount-1
DisplayGrid.Cell(DisplayGrid.ListCount-1, i) = Trim(fields(i)) 'displays the
data
read from file
FileReader.Table(i,DisplayGrid.ListCount-1) = CDbl(Trim(fields(i))) 'and
puts the values into an array
next
wend
FileReader.DataLines=DisplayGrid.ListCount-1 'no of lines stored in Table array
'close file
tis.close
End Sub
```

FileReader Control Display:

```
Sub Action()
'FileReader properties
'Offset1, Offset2 As Double - Load bar zero offsets calculated here
'Threshold As Double - calculate threshold for animal presence here
'FilePosition As Integer - current line in file for starting next scan
'Table(5,150000) As Double - array holding 1 hours WOW data
'DataLines As Integer - number of valid data lines in Table
'EarTag As Integer - last read ear tag value which may have to transfer from one
cow
to the next
'EarTagFlag As Boolean - an indicator that there is a current ear tag
'output data to second window
dim line as integer
dim column as integer
For line = 0 to FileReader.DataLInes
OutPutGrid.AddRow""
for column = 0 to OutPutGrid.ColumnCount-1
OutPutGrid.cell(line,column) = cstr(FileReader.Table(column,line))
next
next
End Sub
```

FileReader Control Initialise:

```
Sub Action()
'FileReader properties
'Offset1, Offset2 As Double - Load bar zero offsets calculated here
'Threshold As Double - calculate threshold for animal presence here
'FilePosition As Integer - current line in file for starting next scan
'Table(5,150000) As Double - array holding 1 hours WOW data
'DataLines As Integer - number of valid data lines in Table
'EarTag As Integer - last read ear tag value which may have to transfer from one
cow
to the next
'EarTagFlag As Boolean - an indicator that there is a current ear tag
'TagMax and TagMin As Double - Range of Valid Ear Tags
'WeighLine As Integer
Dim Bar1 as Double
Dim Bar2 As Double
Dim Load As Double
Dim TotalWeightData As Double
Dim NValidWeights As Integer
```

```
Dim Line As Integer
Dim Column As Integer
'set threshold
FileReader.Threshold=2800
'set valid ear tags
FileReader.TagMax=4000000
FileReader.TagMin=3000000
'calculate offsets by averaging first 30 values in file
TotalWeightData=0 'initialise
NValidWeights=0
Bar1=0
Bar2=0
for NValidWeights = 0 to 29
Bar1 = Bar1 + FileReader.Table(3,NValidWeights)
Bar2 = Bar2 + FileReader.Table(4,NValidWeights)
next
FileReader.Offset1 = Bar1/30.0
FileReader.Offset2 = Bar2/30.0
'Initialise other variables for start of weight calculations
FileReader.FilePosition = 0
FileReader.EarTag = 0
FileReader.EarTagFlag = false
FileReader.WeighLine=0
'Display calculated offsets and other variables
For line = 0 to 100000
OutPutGrid.AddRow""
for column = 0 to 5
OutPutGrid.cell(line,column) = ""
next
next
OutPutGrid.cell(0,0) = "Offset 1"
OutPutGrid.cell(0,1) = cstr(FileReader.Offset1)
OutPutGrid.cell(1,0) = "Offset 2"
OutPutGrid.cell(1,1) = cstr(FileReader.Offset2)
OutPutGrid.cell(3,0) = "Data lines"
OutPutGrid.cell(3,1) = cstr(FileReader.DataLines)
FileReader.Line = 5
End Sub
```

FileReader Control Weight:

```
Sub Action()
'FileReader properties
'Offset1, Offset2 As Double - Load bar zero offsets calculated here
'Threshold As Double - calculate threshold for animal presence here
'FilePosition As Integer - current line in file for starting next scan
'Table(5,150000) As Double - array holding 1 hours WOW data
'DataLines As Integer - number of valid data lines in Table
'EarTag As Integer - last read ear tag value which may have to transfer from one
cow
to the next
'EarTagFlag As Boolean - an indicator that there is a current ear tag
'EarTagPosn As Integer
'Line and Column as integers
'WeighLine as Integer
Dim Weight As Double
Dim EarTag As Integer
Dim TotalWeight As Double
Dim ThisWeight As Double
Dim NewThreshold As Double
Dim NValues As Integer
Dim StartPosn As Integer
Dim EndPosn As Integer
Dim ValidRange As Boolean
Dim ValidStart As Boolean
Dim Less As Boolean
Dim EOF As Boolean
Dim ijk As Integer
```

```
' Start search for next animal
TotalWeight=0
NValues=0
EOF = False
ValidStart = False
ValidRange= False
Less = True
'read through file to find valid start and end points above threshold
'to be valid it is required that
'(i) an eartag is identified prior to the upward crossing;
'(ii) the range between the start and end points is greater than X; and
'(iii) the end of the file is not reached.
'If (i) or (ii) are not satisfied, then skip entire above threshold period, but
retain and
update
'any eartag data.
'If (iii) is not satisfied, then exit.
While not (ValidStart and ValidRange) and (not EOF)
'is there an eartag here; if so, save details
ijk =FileReader.Table(5,FileReader.FilePosition)
If (ijk>FileReader.TagMin) and (ijk<FileReader.TagMax) then
FileReader.EarTagPosn=FileReader.FilePosition
FileReader.EarTag=ijk
FileReader.EarTagFlag=True
end if
Weight= (FileReader.Table(3,FileReader.FilePosition) - FileReader.Offset1 +
FileReader.Table(4,FileReader.FilePosition) - FileReader.Offset2)/2.0
'is there an upward crossing of the threshold
if Less and (Weight>=FileReader.Threshold) then
Less=False
if FileReader.EarTagFlag then
ValidStart=True
StartPosn = FileReader.FilePosition
EarTag= FileReader.EarTag
end if
end if
'is there a downward crossing of the threshold
if not Less and (Weight < FileReader.Threshold) then
Less=True
if ValidStart then
EndPosn=FileReader.FilePosition-1
If NValues >=10 then
ValidRange=True
if FileReader.EarTagPosn<StartPosn then FileReader.EarTagFlag = False
'debug write
OutPutGrid.cell(FileReader.line,0) = "Valid"
OutPutGrid.cell(FileReader.line,1) = cstr(EarTag)
FileReader.Line=FileReader.Line+1
'end debug write
else
'debug write
OutPutGrid.cell(FileReader.line,0) = "Invalid"
OutPutGrid.cell(FileReader.line,1) = cstr(StartPosn)
OutPutGrid.cell(FileReader.line,3) = cstr(EndPosn)
OutPutGrid.cell(FileReader.line,4) = cstr(EarTag)
FileReader.Line=FileReader.Line+1
'end debug write
ValidStart=False
TotalWeight=0
NValues=0
end if
end if
end if
'if above threshold then accumulate data
if (not Less) and ValidStart then
TotalWeight=TotalWeight+Weight
```

```
NValues=NValues+1
end if
'increment file pointer and check for end of file
FileReader.FilePosition=FileReader.FilePosition+1
if FileReader.FilePosition >=FileReader.DataLines then EOF = True
wend
'Now do first pass of weight calculation
If not EOF then
'Calculate mean weight and if mean different from threshold then move threshold
to mean and find new start and end points etc
ThisWeight=TotalWeight/NValues
'debug write
OutPutGrid.cell(FileReader.line,0) = "Begin:"
OutPutGrid.cell(FileReader.line,1) = cstr(StartPosn)
OutPutGrid.cell(FileReader.line,3) = cstr(EndPosn)
OutPutGrid.cell(FileReader.line,4) = cstr(ThisWeight)
FileReader.Line=FileReader.Line+1
'end debug write
NewThreshold=FileReader.Threshold
While (ThisWeight > (NewThreshold+1))
NewThreshold=ThisWeight
'move start position forward
Weight = (FileReader.Table(3,StartPosn) - FileReader.Offset1 + FileReader.Table
(4,StartPosn) - FileReader.Offset2)/2.0
While Weight < NewThreshold
TotalWeight=TotalWeight-Weight
StartPosn=StartPosn+1
NValues=NValues-1
Weight = (FileReader.Table(3,StartPosn) - FileReader.Offset1 +
FileReader.Table(4,StartPosn) - FileReader.Offset2)/2.0
Wend
'move end position back
Weight = (FileReader.Table(3,EndPosn) - FileReader.Offset1 + FileReader.Table
(4,EndPosn) - FileReader.Offset2)/2.0
While Weight < NewThreshold
TotalWeight=TotalWeight-Weight
EndPosn=EndPosn-1
NValues=NValues-1
Weight = (FileReader.Table(3,EndPosn) - FileReader.Offset1 + FileReader.Table
(4,EndPosn) - FileReader.Offset2)/2.0
Wend
ThisWeight=TotalWeight/NValues
'debug write
OutPutGrid.cell(FileReader.line,0) = "Cont:"
OutPutGrid.cell(FileReader.line,1) = cstr(StartPosn)
OutPutGrid.cell(FileReader.line,3) = cstr(EndPosn)
OutPutGrid.cell(FileReader.line,4) = cstr(ThisWeight)
FileReader.Line=FileReader.Line+1
'end debug write
Wend
'Have now calculated a weight with corresponding eartag, so output these values
WeightGrid.AddRow""
WeightGrid.cell(FileReader.WeighLine,0)=cstr(EarTag)
WeightGrid.cell(FileReader.WeighLine,1) =cstr(ThisWeight/10.0)
FileReader.WeighLine=FileReader.WeighLine+1
end if
FileReader.Line = FileReader.Line+1
End Sub
End Class
```

Glossary

DairyScale – Scale head. It is a piece of hardware that the dairy cow algorithm runs on, see 3.1.3 DairyScale Scale Head.

EID – Electronic identification number.

PreM – Premature addition to the sum threshold algorithm, running in the code of the CSV application.

RFID – Radio frequency identification. A system that uses radio frequency to transfer data.

WOW – Walk-over weighing. The crate that the weighing platforms are attached to, see 3.1.1 WOW Crate.