Lameness Detection in Sheep Using 3- Axis Accelerometer Sensor

Zainab Al-Rubaye

Berk Morkoc

Supervisors

Ali Al-Sherbaz

Michael Opoku Agyeman

Scott Tuner

UNIVERSITY OF NORTHAMPTON

Department of Computing

Northampton, 2018

Abstract

Monitoring animal behaviour and disease can prove challenging when working in inaccessible environments. This problem can be addressed by using animal attached accelerometers. This study considers finding a feasible method to improve the detection of lameness in sheep under favour of accelerometer placed on the animal. Accelerometer sensor was attached to sheep's neck and programmed by Nodemcu (ESP-8266) microcontroller.

The sensor measurement interval is set to +-2g and sampling frequency is set to 1000Hz. The direction of the movement is associated with the axis and the Y axis was determined as an axis which is related to the up-down movement. In order to detect forward motion, linear acceleration in Y axis was calculated and used on calculation of the velocity. The velocity calculation purpose is to determine when the detection of lameness will be started measuring by the accelerometer sensor.

The data achieved was stored in an excel file through an external programme called as PLX-DAQ. However, only 10 data recorded in 1 sec despite 1000Hz sampling frequency due to the delay in communication between serial port and storage. Accuracy of the velocity calculation is very low because of the sensor drifts and computing errors. Therefore, other numerical integration methods and various filters can be applied in order to reduce calculation errors and sensor drifts.

Table of Contents

I. INTRODUCTION

The study of animal behaviour provides us with a better understanding and insight into their health. While monitoring animals in an overseen environment is relatively simple, this is not true for unsupervised animals in their natural environment. In this latter case, detection and estimation can be achieved by attaching sensors to the animal. Depending on the type of sensor, information regarding the animals energy expenditure, location, speed, heart rate, temperature and behaviour can be collected. This provides researchers with data on the behaviour of animals in environments that are inaccessible for human observation.

Lameness is an unpredictable disease that can cause death unless the sheep are treated. A gait glitch can be observed on the lame sheep, however, detecting this observation by eyes is not possible. The lame sheep abnormally moves it's body up and down while walking. Generally, the farmer needs to check out the sheep's hoofs regularly by catching it. However, many farmers do not do this as it's troublesome. This project aims to find a feasible method to improve the detection of lameness in sheep. This will result in more sheep death prevention and a lower overall cost of sheep treatment.

An accelerometer is often used in animal monitoring systems. It can be placed in a lightweight tag which can be attached to the animal with minimal obstruction or interfering with the natural movement of an animal. Mono-axial and bi-axial accelerometers are available, but have a few disadvantages compared to a tri-axial accelerometer [1].

Measuring the acceleration in all three axes provides a more complete picture of animal movement patterns and could reveal information that would have otherwise been missed.

Several studies have considered the usage of accelerometer but none of them were aimed to identifying lameness in sheep. The previous studies done are related to the classification of sheep behaviours which are grazing, resting and walking. [1] In these classifications, linear discriminant analysis (LDA) and Quadratic discriminant analysis (QDA) were trained based on 10 features and a greedy selection procedure was used to determine which features provide the highest accuracy. The LDA and QDA classifiers achieved an overall accuracy of 87.1% and 89.7% respectively. However, grazing was misclassified, because it was confused with lying.

In another research [2], estimation of the energy expenditure of grazing farm animals was tried to achieve with calculation of overall dynamic body acceleration (ODBA) to the accompaniment of the accelerometer sensor. In addition to this, the relationship between ODBA and heart rate has been observed. In calculation of overall dynamic body acceleration (ODBA) following procedures were used: static acceleration was first approximated by smoothing the obtained acceleration using 2-sec running means, according to Halsey et al. [3]; dynamic acceleration was calculated by subtracting the static acceleration from the raw [7, 8] acceleration; and ODBA was calculated by summing the three spatial absolute dynamic accelerations. Afterwards, it was aimed that estimation of the energy expenditure and its relationship with heart rate.

In the present project, an Mpu-6050 accelerometer which has ability of measuring in 3 Axis, works in conjunction with a Nodemcu (ESP-8266) microcontroller to detect the lameness. The resources shown in the hardware section were used for data collection.

II. MATERIALS AND METHODS

a. HARDWARE

The hardware of the project consists of a 3-axis accelerometer, breadboard and Esp-8266 microcontroller. The tag was designed to allow the accelerometer to be constantly sampled at a frequency of 1000Hz and to subsequently store the measurements in the excel file.

(Fig. 1) The x-axis is associated with the left-right movement of the sheep. The z-axis is associated with the up-down movement of the sheep and the y-axis is associated with the forward-backwards movement of the sheep.

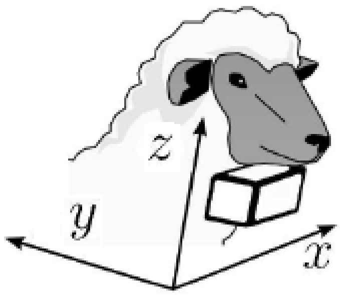

Figure 1: A sketch of the accelerometer axes relative to the sheep

b. DATA COLLECTION

i) Wiring and Installing the Circuit

Nodemcu (ESP-8266) is a microcontroller that enables the encoding of sensors and the processing of the sensor value. A breadboard was used in order to place the Nodemcu micro-controller and Mpu-6050 accelerometer sensor. (1) Following wiring conception below was applied to create the circuit.

$$\begin{gathered} 3V3 \rightarrow VCC \\ GND \rightarrow GND \\ D7 \rightarrow SDA \\ D6 \rightarrow SCL \end{gathered} \tag{1}$$

(Fig. 2) The *3V3* pin was used as positive voltage and the *GND* pin was used as negative voltage in the installation of the circuit. *D7* and *D6* pin are digital pins on Nodemcu microcontroller and these two pins were used on communication between microcontroller and accelerometer sensor.

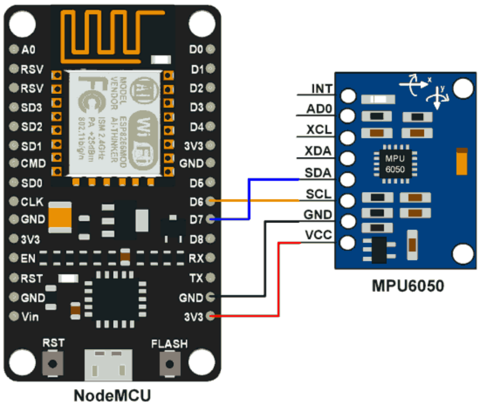

Figure 2: Wiring schema of conception shown (1)

ii. Necessary Programs and Drivers

Nodemcu drivers were set up and a program called as Arduino IDE was used to encode the Nodemcu microcontroller. Afterwards, Nodemcu was connected to computer by USB cable. Thereby, the circuit section was completed with it.

iii. Sensor Data Collection

The following codes below were used to get the sensor data as a raw data *(Fig. 3).* In order to convert the raw data to g unit, a scale factor that refers the measurement of accelerometer is $+-2g$ has been encoded. The sensor data calibrated when it is held parallel to the ground and standing on the straight floor is adjusted as $X=0g$, $Y=0g$ and $Z=1g$ [6].

```
const char * ssid = "iPhone";

const char * password = "morkocberk";

// MPU6050 Slave Device Address
const uint8_t MPU6050SlaveAddress = 0x68;

// Select SDA and SCL pins for I2C communication
const uint8_t scl = D1;
const uint8_t sda = D2;



// sensitivity scale factor respective to full scale setting provided in datasheet
const uint16_t AccelScaleFactor = 16384;
const uint16_t GyroScaleFactor = 131;

// read all 14 register
void Read_RawValue(uint8_t deviceAddress, uint8_t regAddress){
Wire.beginTransmission(deviceAddress);
Wire.write(regAddress);
Wire.endTransmission();
Wire.requestFrom(deviceAddress, (uint8_t)14);
AccelX = (((int16_t)Wire.read() << 8) | Wire.read());
AccelY = (((int16_t)Wire.read() << 8) | Wire.read());
AccelZ = (((int16_t)Wire.read() << 8) | Wire.read());
Temperature = (((int16_t)Wire.read() << 8) | Wire.read());
GyroX = (((int16_t)Wire.read() << 8) | Wire.read());
GyroY = (((int16_t)Wire.read() << 8) | Wire.read());
GyroZ = (((int16_t)Wire.read() << 8) | Wire.read());
}
if(millis() - lastTime > 1){ //sample in 1msec

Read_RawValue(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_XOUT_H);

//divide each with their sensitivity scale factor
Ax = (double)AccelX/AccelScaleFactor;
Ay = (double)AccelY/AccelScaleFactor;
Az = (double)AccelZ/AccelScaleFactor;
T = (double)Temperature/340 + 36.53; //temperature formula
Gx = (double)GyroX/GyroScaleFactor;
Gy = (double)GyroY/GyroScaleFactor;
Gz = (double)GyroZ/GyroScaleFactor;
```

Figure 3: Codes written for getting raw accelerometer data. Sampling frequency is set 1ms which is equal to 1000Hz. Scale factors refer that the sensor measurement interval +-2g.

iv. Measurement of Linear Acceleration and Velocity

In calculation of linear acceleration following formula was used:

$$g = 0.9 * g + 0.1 * a \qquad (2)$$

$$a = a - g$$

where g is the earth gravity and a is the acceleration value of Y axis *(2)*. The purpose of the calculation of the linear acceleration is to remove the earth gravity from the original acceleration data that is measured by the Mpu-6050 accelerometer. Since, the sensor will not be stand, linear acceleration calculation is mandatory to get the accurate velocity to detect whether the sheep is moving or not.

Regarding the calculation of velocity was calculated the following code block below.

```
aym = aycopy/16384;
totalAy+= aym; //add each meausured Ay value
counter ++;
if(counter == 10){
 average = (double)totalAy/10;
 counter = 0;
 totalAy = 0;
 v = v0 + (average * 9.8 * 0.1); //v0 = 0 and a0 = 0 in the beginning
 v0 = v; // given the previous v value to v0 for the next loop
 a0 = average;
 v = fabs(v);
}
lastTime = millis();
}
```

Figure 4: Calculation of velocity

According to codes above *(Fig. 4)*, *aycopy* value is diveded by *16384* which is scale factor to exchange the raw value. Afterwards, the values are being added to each other and counter is increased. The purpose of doing that is to have an average of acceleration Y value for each 10 data achieved.

Following formula is used for calculation of the velocity:

$$V = V0 + at \qquad (3)$$

Where $v0$ is the first velocity, a is acceleration in Y axis and t is time (3). However, when the codes reviewed, you can see that acceleration is multiplied by 9.8. The reason of doing that is sensor values are in terms of g unit. $(\mathrm{g} = 9.8\ \mathrm{m}/s^2)$

v. Data Storage

Since, Arduino IDE is not able to forward the data directly from serial port to the computer, another interface program called PLX-DAQ which provides us to get the values seen on the serial port was adopted for the project.

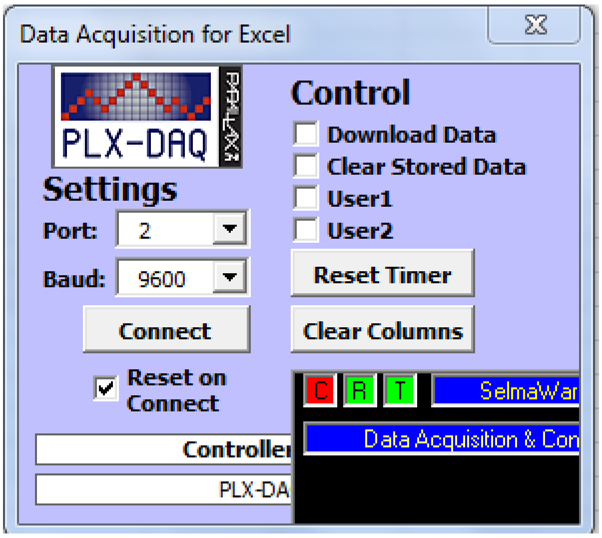

Figure 5: A screen shot from the program used for data transfer to excel.

This programme creates its own excel file and seems extension when the excel is opened. After opening the programme, all needs to be set is to select the correct port and baud rate for communication.

III. RESULTS AND DISCUSSIONS

Time, Acceleration X, Acceleration Y, Acceleration Z, Linear Acceleration Y and Velocity respectively can be seen from the schema below. (Fig. 6)

Time	Acceleration X (g)	Acceleration Y (g)	Acceleration Z(g)	Linear Acceleration Y(g)	Velocity (m/s)
15:07:25	-0.19	-0.01	1.02	0	0.03
15:07:25	-0.19	-0.02	1.00	0	0.03
15:07:25	-0.17	0	1.02	0,01	0,04
15:07:25	-0.17	-0.01	1.02	0	0,03
15:07:25	-0.18	-0.01	1.02	-0.02	0,03
15:07:26	-0.18	-0.02	1.02	-0.02	0,14
15:07:26	-0.17	0	1.03	0.01	0.06
15:07:26	-0.28	0.04	1	0.01	0.06
15:07:26	-0.17	0	1.03	0.01	0.02
15:07:26	-0.23	0.01	1.06	0.01	0.09
15:07:26	-0.28	0.03	1.13	0	0.17
15:07:26	-0.46	-0,06	1.11	-0.01	0.46
15:07:26	-0.61	-0,18	0.88	-0.02	0.87
15:07:26	-0.62	-0.34	0.5	-0.03	1.3
15:07:26	-0.78	-0,42	0.32	-0.01	1.39
15:07:27	-0.81	-0,44	0.27	0	1.41
15:07:27	-0.83	-0,44	0.28	0	1.38
15:07:27	-0.86	-0,44	0.31	0	1.38
15:07:27	-0.84	-0.44	0.28	0	1.38

Figure 6: Data collection schema. Time, acceleration X, acceleration Y, acceleration Z, linear acceleration Y and velocity, respectively. These data transferred from the excel file. Sampling frequency is 1000Hz, however, frequency of achieved data to excel file is 10Hz due to delay.

There are many numerical integration techniques such as Simpson, Trapeziodal (Newton) can be applied to calculate the velocity from acceleration [4, 5]. In the present study, a different comprehension was adopted to calculate velocity. Formula of velocity calculation from the acceleration value (3) was used to escape the numerical integration error, however, this computing adaption caused new calculation errors.

Accuracy of measured velocity is very low because of the calculation mistakes and sensor drifts. The only way to calculate the velocity from the acceleration is integration. Since, unknown functions exist, integration of the values achieved always includes calculation errors. That's why, these small integration mistakes cause big deviations from the real value by time passes.

IV. CONCLUSION

Reviewing animal's behaviour gives us an opportunity to understand their natural life. In the present project, a disease called lameness in sheep has been tried to detect. Lameness is a sick which is related to sheep movement type. The lame sheep moves abnormally in comparison with the healthy sheep. The most recognisable symptom of the lameness is irregularly up and down movement of the sheep. However, it is very difficult to estimate it by eyes. Therefore, specialized devices should be used for the detection. Last but not least about the disease, if the illness is not treated, it cause of sheep death.

In this work, Mpu-6050 3 - Axis accelerometer sensor was used in order to detect the lame in sheep. It is determined to place the sensor on the neck of the sheep. Thus, detection of lame will be done, using accelerometer acceleration data. Before start detecting the lame, it is necessary to deduce that the sheep is walking, running or standing. The only way to detect the lame accurately is to understand movement because the illness can be merely obtained while the sheep is moving. For this reason, velocity was tried to compute with acceleration data. Y – Axis acceleration data was associated with forward backward movement. In calculation of the velocity, a formula was used and sampling frequency was set to 1000Hz. Nevertheless, only 10 data were stored in the excel file due to communication delay.

Accuracy of the velocity calculation is very low because of the sensor drifts and calculation errors. So, this research showed that the sensor drifts need to be reduced by using a filter and in order to achieve accurate velocity value, the calculation method needs to be altered. In addition, any other auxiliary sensors such as GPS tracking, motion detector can be attached to the present work.

In the future, new studies related will be based on this research. The calculation errors will be reduced, implementing a numerical integration method. Afterwards, the sensor will be put on the sheep and real data will be collected to an excel file through Wi-Fi.

V. REFERENCES

1. Automatic classification of sheep behaviour using 3-axis accelerometer data
2. Jacques Marais, Solomon Petrus Le Roux, Riaan Wolhuter and Thomas Niesler Department of Electrical and Electronic Engineering Stellenbosch University, South Africa

3. Application of Overall Dynamic Body Acceleration as a Proxy for Estimating the Energy Expenditure of Grazing Farm Animals: Relationship with Heart Rate. Yan Ropert-Coudert, Academic Editor

4. Halsey LG, Shepard ELC, Quintana F, Laich AG, Green JA, Wilson RP. The relationship between oxygen consumption and body acceleration in a range of species. Comp Biochem Physiol A Mol Integr Physiol. 2009; 152: 197–202. oi: 10.1016/j.cbpa.2008.09.021 [PubMed]
5. [Numerical Integration Formulas, Berlin Chen, Department of Computer Science& Information Engineering, National Taiwan Normal University
6. Applied Numerical Methods with MATLAB for Engineers, Chapter 19 & Teaching material
7. Use of a tri-axial accelerometer for automated recording and classification of goats′ grazing behaviour Mae¨g Moreau a, Stefan Siebert b, Andreas Buerkert c, Eva Schlecht a

8. Shepard ELC, Wilson RP, Halsey LG, Quintana F, Laich AG, Gleiss AC, et al. Derivation of body motion via appropriate smoothing of acceleration data. Aquat Biol. 2008; 4: 235–241.
9. Gleiss AC, Wilson RP, Shepard ELC. Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure. Methods Ecol Evol. 2011; 2: 23–33.

10. https://www.youtube.com/watch?v=m_Te1K_ZTJo&index=18&list=PLDRcccSktQd6zXC4_ri_xxMsdgrH5MCPS
11. http://www.electronicwings.com/nodemcu/mpu6050-interfacing-with-nodemcu
12. https://firebase.google.com/?gclid=EAIaIQobChMIpuCCx5Pr3AIVhZztCh0suwhJEAAYASAAEgKh2PD_BwE
13. https://www.instructables.com/id/Controlling-an-NodeMCU-ESP8266-10-12E-With-Google-/
14. https://yakutozcan.blogspot.com/2017/11/nodemcu-firebase-ve-android-ile-led-uygulamasi.html
15. https://github.com/