Continuous Auscultation in the Wild

Shyam A. Tailor
Downing College

A dissertation submitted to the University of Cambridge in partial fulfilment of the requirements for the degree of Master of Engineering in Computer Science

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: sat62@cam.ac.uk

May 30, 2019

Declaration

I Shyam A. Tailor of Downing College, being a candidate for Part III of the Computer Science Tripos, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 11904 (calculated using `texcount`)

Signed:

Date:

Acknowledgements

I would like to thank my supervisor, Cecilia, for her advice and support during this project, along with all the other members of the Mobile Systems Group. I am also grateful to Brian Jones for his electronics advice.

Abstract

Non-speech body sounds collected from the human body have many medical uses: several respiratory and cardiovascular diseases are diagnosed using audio data. In recent years, wearable devices for fitness and health tracking have become popular, but no commercially available device tracks audio data collected from the body, as there remain many technical challenges. This dissertation explores the viability of building a chest-mounted wearable device that can be used for continuous health monitoring. A custom device was designed and manufactured, and was subsequently used to collect a data set from 9 individuals. The user study focuses on exploring the noise tolerance of a wearable device; unlike related work, this study explicitly considered robustness to ambient noise and user motion.

Two algorithms were proposed for continuous heart monitoring: a autocorrelation-based technique that yielded an estimate of the heart rate, and a more complicated technique which segments the collected audio into the different phases of the cardiac cycle. Both techniques yielded accurate heart rate estimates when the user was resting: 1.86% and $0.26 \pm 0.02\%$ median percentage errors were found for the two algorithms respectively, even under challenging ambient noise conditions. The segmentation algorithm yielded good estimates even when the user was walking, and could also be used to obtain an accurate measure of heart rate variability. Both algorithms were also evaluated by considering the viability of running them on-device; it was shown that the cheaper algorithm could run continuously for over a week on a typical battery found in a wearable.

Finally, the viability of using a device for continuous asthma symptom detection was considered. An approach employing convolutional neural networks running on-device was assessed. It was found that the model could obtain near-human level performance at detecting wheezing from audio continuously, with a battery life of over 3 days.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years there has been an explosion in the number of wearables offering fitness or medical features [71]. This market extends beyond devices marketed towards consumers, such as fitness trackers by Fitbit, or smartwatches such as the Apple Watch. A growing part of the market are devices sold to medical professionals which are prescribed to patients to enable continuous health monitoring. Despite there being significant diversity in the wearables market, with devices varying in body placement and included sensors, a device which continuously monitors the wearer's *body sounds* remains elusive. Auscultation, the practice of listening to body sounds, is a diagnostic technique that has been used for at least two centuries [57]. It can be used to diagnose several circulatory, respiratory and digestive conditions, many of which cannot be diagnosed using sensing modalities that are usually incorporated into wearable devices.

There has been decades of research into speech processing: voice assistants, such as Apple's Siri [53] or Amazon's Alexa [2], are available to billions of people worldwide. It is natural to wonder whether it is possible to use *non-speech* body sounds, such as heart, breathing, or digestive sounds, for fitness and medical purposes.

There are substantial challenges that must be considered in order to exploit audio collected from the body in a wearable device. One challenge is dealing with the

increase in data that must be processed, in comparison to the quantity obtained from accelerometers or gyroscopes. While it would be common to sample an accelerometer at $\mathcal{O}(10^2)$Hz, common audio sampling rates are $\mathcal{O}(10^4)$Hz, depending on the precise application. As wearable devices are constrained in both computation and energy consumption, it is necessary to build techniques that can scale to greater quantities of data. Another challenge faced by wearables is that they are used in the wild: algorithms used must tolerate noisy measurements. Finally, there are concerns regarding collecting audio due to its sensitive nature. Privacy is a worry for potential users: voice assistants often upload data to the cloud to perform inference, stoking these concerns. Inference accuracy must be balanced against energy efficiency and privacy.

With these challenges in mind, this dissertation makes the following contributions:

1. The construction of a novel device for collecting heart and respiratory sounds from the body. The device is *discreet* and its design explicitly considers noisy conditions.

2. The collection of a new dataset which assesses the impact of real world conditions on the device. The user study considers high ambient noise levels, and the effect of user activities, including motion. Prior work does not adequately assess these conditions.

3. The assessment of two novel algorithms proposed for continuous heart monitoring while subject to these noisy conditions. The first algorithm allows for the heart rate to be estimated cheaply, while the second algorithm yields more accurate estimates for the heart rate, along with heart rate variability information. Both algorithms can be run *on-device*, and it was found that the cheaper algorithm could be run in real-time with a battery life of over a week on plausible hardware.

4. The development of a technique for monitoring asthma symptoms in real time. This algorithm can also be run on-device, while obtaining battery life of over 3 days on plausible hardware.

Chapter 2

Background

This chapter begins by explaining the value of medical wearables, before moving to discussing related work. Finally, necessary biological background is provided.

2.1 Context

Wearable devices that continually monitor the user's health allow for detecting the onset of a new illness with lower latency, and continuous tracking of symptoms of illnesses the wearer already has. Accurate and effective wearable medical devices would mean that healthcare is no longer *reactionary*, but instead *preventative*: medical professionals could be informed of worsening patient health, even when the patient is not in a clinical setting. There are several benefits to this paradigm shift: earlier patient interventions would lead to improved outcomes, and lower costs [71].

A concrete medical application of a device would be to monitor asthma, which kills hundreds of thousands of people each year [15]. A wearable device could detect asthma symptoms before the user is even aware of an issue and prompt them to take action, before it is too late.

2.2 Related Work

2.2.1 Existing Devices

The idea of monitoring body sounds with a wearable has been explored [49, 34], but the devices proposed do not address energy efficiency or noise robustness. Another challenge is that these devices are not *practically* wearable: BodyBeat [49] was worn on the neck. In reality wearable devices are not used if they inconvenience the user in any way: even life-saving wearable defibrillators were not worn by patients they were prescribed to in a recent study [44]. Another issue with BodyBeat was that inference was performed by streaming all data to a smartphone. In recent years iOS and Android have introduced aggressive measures to maximise battery life, and this method is no longer practical. Inference must be performed *on-device*.

2.2.2 Body Sound Classification

A recording of heart sounds is known as a phonocardiogram (PCG), and techniques for automated analysis have been investigated. Traditionally, efforts have applied standard digital signal processing techniques, such as autocorrelation, to extract information from the PCG [55]; recent efforts have involved deep learning techniques [65, 51].

Automated analysis of ventilatory noise has also been studied. One problem that has been investigated is automated cough detection, which allows for the tracking of a symptom of many respiratory diseases [33]. There has also been work into automated detection of the wheezing sounds associated with asthma [34].

In existing work the algorithms developed are not usually analysed in the context of a limited computational and energy budget, but this is a major consideration for a wearable device.

2.2.3 Other Applications of Sound

A recent paper studied how the wearer's internal body voice can be used to identify them [35]. There is also evidence that features in the cardiac cycle can identify individuals [64]. It has been shown that an individual's emotional state [48, 32], stress level [36, 32], and symptoms for sleep apnea [42] can be detected through recordings made by a smartphone. Finally, vocal features from recordings have been used to diagnose bipolar disorder [13] and post-traumatic stress disorder [4].

2.2.4 Machine Learning Inference on Resource Constrained Devices

Approaches to improving efficiency on smartphone-class devices have included model compression, quantising model weights and pruning unimportant weights [31, 56]. These devices are constrained in terms of storage and computational resources; this is especially true when moving to microcontrollers.

It is now viable to deploy neural network models leveraging to smartphones [59, 12]. At time of writing machine vision tasks are well supported, but it is difficult to use recurrent networks [59]. The research community has proceeded to exploring the viability of performing inference on embedded processors [31].

2.3 Body Sounds During Auscultation

There are several sources of sounds that a physician will listen to with a stethoscope. One source is the heart. It is also common to listen to noises made by the respiratory system during breathing [28]. "Crackles" are an abnormal sound associated with pneumonia and other illnesses, such as cystic fibrosis, which cause a fluid build-up in the lungs. "Wheezes", however, are associated with asthma and other illnesses which cause the airways to constrict. Bowel sounds are also

monitored by medical professionals. Rather than listening for the sounds, clinicians are interested in their absence: a lack of bowel sounds is a sign of an ileus, which occurs if there is a loss of muscular activity in the bowel [3].

2.3.1 The Cardiac Cycle

As this dissertation will explore algorithms for extracting information from heart sounds, it is worth understanding how the sounds arise. The heart consists of four chambers: two atria, and two ventricles, shown in fig. 2.1a [16]. The atria force blood into the ventricles, which then pump the blood into the arteries. There are two pairs of valves in the heart: the atrioventricular (AV) valves, also known as the mitral and tricuspid valves, which allow blood to flow from the atria to the ventricles. The semilunar valves, also known as the aortic and pulmonary valves, allow blood to flow from the ventricles into the arteries. The cardiac cycle proceeds as follows:

1. AV and semilunar valves are shut; blood flows from into the atria from the veins.

2a. AV valves open, and blood starts to flow from the atria into the ventricles.

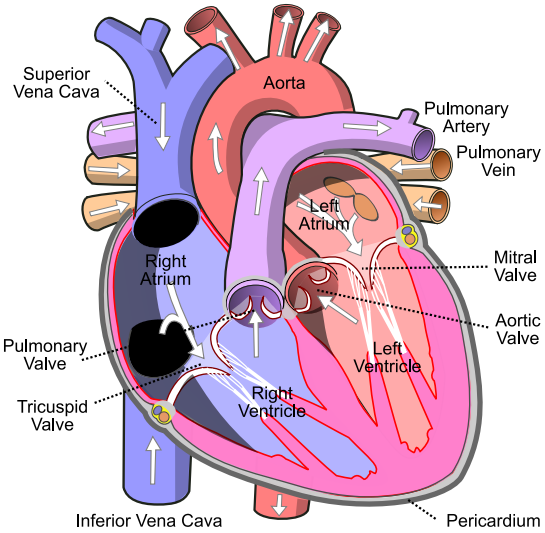
2b. Atria contract to force blood into the ventricles.

3. AV valves shut.

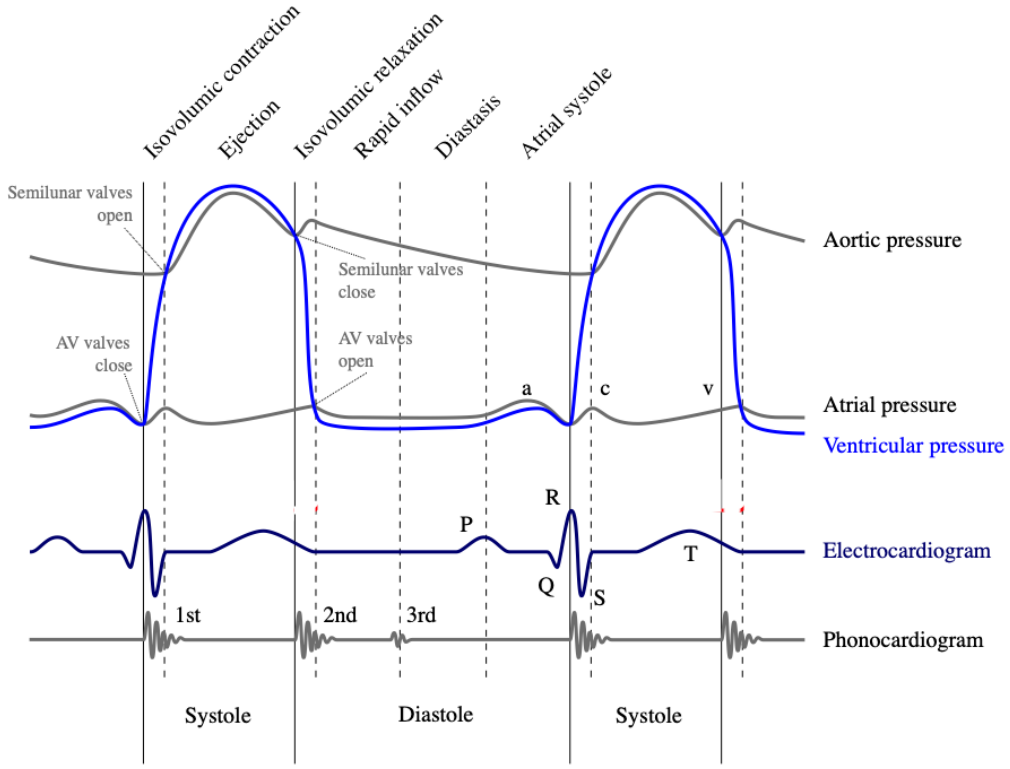
4. Ventricles contract, forcing the semilunar valves to open and let blood into the arteries.

The first two phases are known as diastole, where the heart is relaxed. The last two phases form the systole period, where the heart is beating.

When listening to a healthy heart, two distinct sounds can be heart; these correspond to valve closures. The first sound, S_1, referred to as "lub", occurs at the start of systole when the AV valves shut. The second sound, S_2 ("dub"), corresponds to the semilunar valves shutting, and is usually quieter than the S_1; unlike the S_1, it can split by up to 20-30ms when inhaling. A diagram explaining this is given in fig. 2.1b.



(a) Diagram of the human heart [66]



(b) Wiggers diagram, indicating relation of heart sounds to the cardiac cycle [69]

Figure 2.1: Heart physiology diagrams.

Heart Defects

This dissertation does not evaluate techniques for abnormal heart sound detection due to the difficulty of obtaining a data set. However, it is worth noting deviations from the normal "lub-dub" sounds could be detected, in principle, by a device which can accurately localise the S_1 and S_2 sounds.

Heart murmurs are due to turbulent blood flow in the heart, and they cannot be diagnosed using electrical activity alone: sound is one method to diagnose these abnormalities. It is possible to deduce that an artery has narrowed, or that blood is flowing the wrong way through valves. Examples of different murmur sounds are given in fig. 2.2.

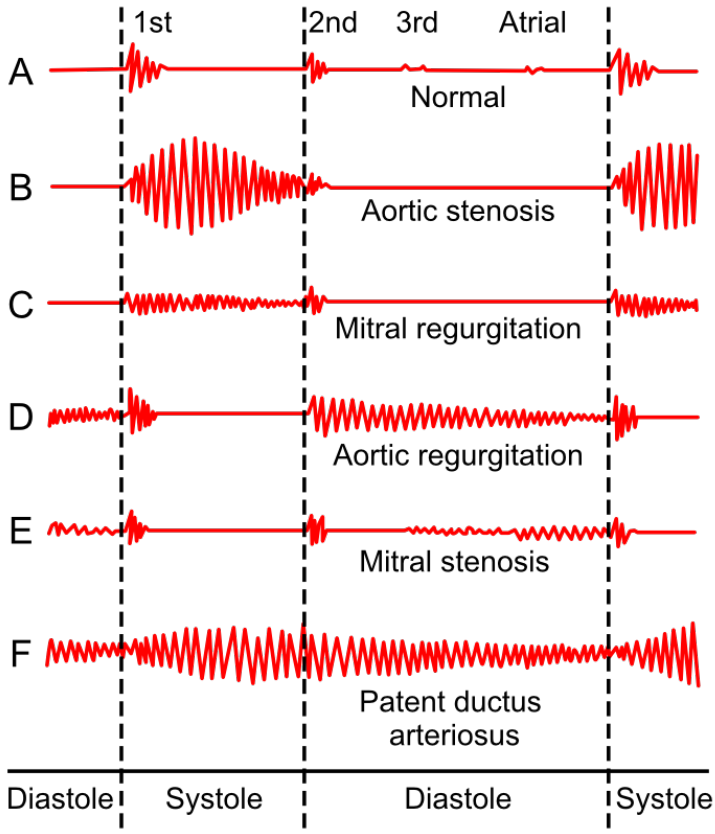


Figure 2.2: Different types of heart murmurs [38].

Chapter 3

Wearable Design and Implementation

This chapter describes the initial experiments used to inform the main requirements for the wearable. The wearable's implementation is described, along with an evaluation of its capabilities. This device is explicitly designed to be discreet and robust to ambient noise.

3.1 Transducer Choice

Fundamentally important to this research is the choice of microphone. The choice is informed by the following factors:

1. Sensitivity to the frequencies of interest. Some of these frequencies are marked on fig. 3.1; note that heart and lung sounds have substantial energy at near-infrasonic frequencies.
2. Robustness to ambient noise.
3. Robustness to friction noise and body motion.

Most electronic devices use electret or condenser microphones. These type of microphones have been used in related devices [70, 34] and are easy to work

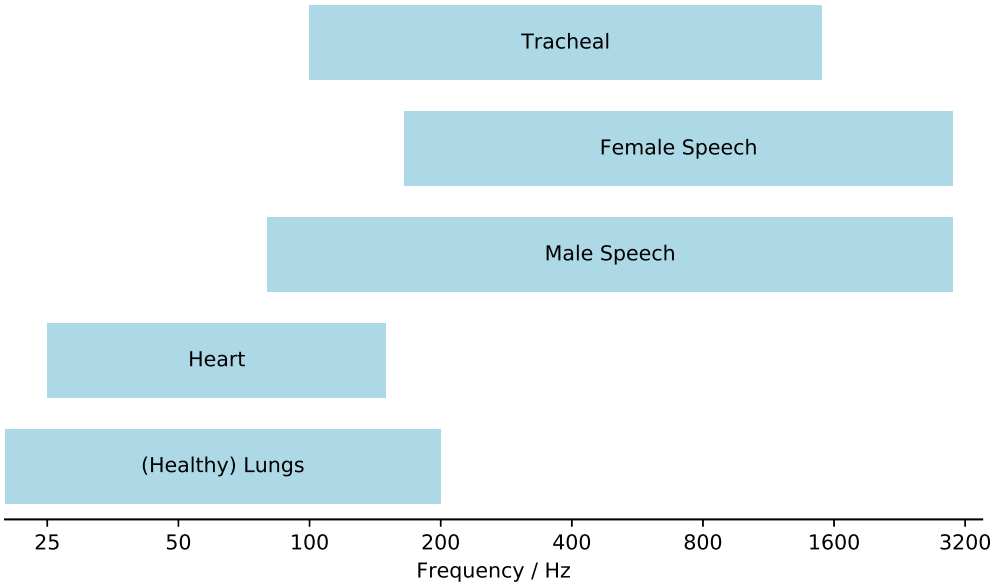


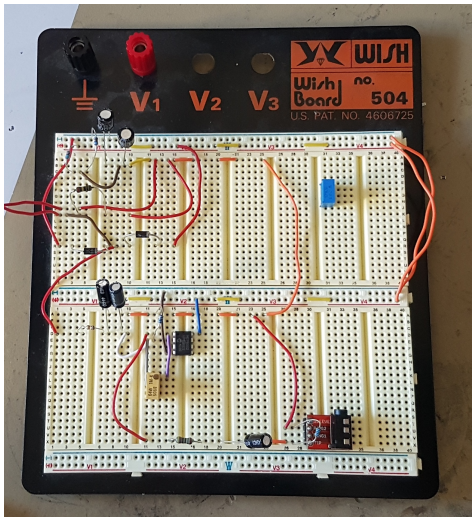
Figure 3.1: Frequency ranges for different body sounds.

with from the perspective of electronics design. However, they do not meet the three criteria. Microphones of these types have poor frequency response below 50Hz and work by picking up vibrations in the air, making them susceptible to ambient noise. It is also likely that these microphones would be susceptible to friction noise during noise.

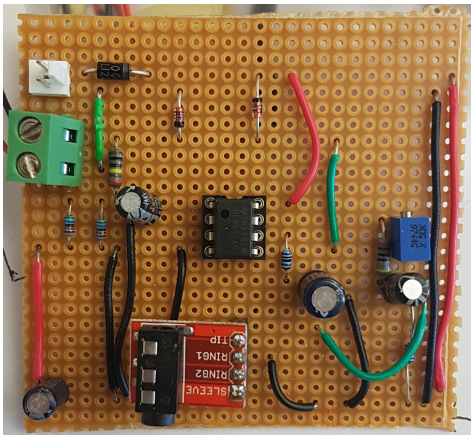
Contact microphones, usually constructed from piezoelectric elements, do not have these issues, and have been used successfully by a related device [49]. They have excellent low-frequency response and pick up vibrations from the skin directly, making them less susceptible to ambient noise. However, these microphones also introduce their own challenges:

1. "Tinny" sound: standard acoustic circuits are not designed for piezoelectric elements which are capacitive. If the amplification circuitry has too low an impedance then the signal will be high-passed, hence it would not be possible to detect heart or lung sounds.

2. Acoustic impedance matching between the skin and microphone: significant mismatch causes most of the signal to be reflected at the boundary.



(a) Breadboard



(b) Veroboard

Figure 3.2: Photos of the circuits used for prototyping.

3.1.1 Initial Experiments

Based on the trade-offs it was decided that contact microphones were the best option for real-world usage. Prototype amplifier circuits (described in more depth in section 3.3.2) were constructed to evaluate different configurations; photos of these circuits can be seen in fig. 3.2.

The first sensor element considered was a brass piezo disk. Application of the disk to the skin could extract a faint heartbeat signal when the piezo element was placed on top of the heart; however, frequencies above 50Hz were significantly attenuated. To improve the acoustic matching, a film of water based gel[1] was placed between the sensor and skin; this improved signal quality, but not to an acceptable level. This setup was also impractical as the gel would dry out.

A piezo film sensor was also evaluated. Taping the film directly to the skin did not provide any useful signal above 5Hz. The respiration waveform, however, could be cleanly extracted if the film was applied vertically on the bottom rib. In this form it is acting as a strain gauge rather than as a microphone; it is already known that strain sensors can recover respiration rate and volume [8].

[1] same as used during medical ultrasound procedures

3.1.2 Chosen Microphone

While it might be possible to build a microphone using a piezoelectric disk, the design space is too large to empirically evaluate: thousands of materials could be used to create a matching layer. Simulating the microphone properties using finite element analysis software, such as OnScale [21], is the most sensible method for finding a matching layer. This work is beyond the scope of this project, and is left as future work.

An off-the-shelf contact microphone [20] designed for integration into electronic stethoscopes was chosen for use in the prototype. An independent study [5] verified that the sensor could be used to record heart sounds. A downside with this sensor was that it required pressure against the skin in order to make a recording, necessitating the use of an elastic strap to hold it against the skin; it was not possible to use (medical) tape, or a patch, to hold the sensor onto the body.

3.2 Identifying Wearable Requirements

Requirements for the proposed wearable were identified, and are given with their motivation.

1. The device must be small enough to be worn on the body, underneath clothing. It is unethical to expect experimental participants to perform the exercises topless.

2. The device must be standalone. One reason for this criteria is that it is important to obtain data when the user is moving. A consistent issue with prior work is that the evaluation does not adequately assess the impact of user motion. Another reason is that piezo signals rapidly degrade with wire length. Charge amplifiers can mitigate these issues to a limited extent, but they are ineffective for lengths of multiple metres.

3. Power consumption needs to be low enough that the device can be run

from a small battery. Larger batteries are heavier meaning that the device will be less stable on the body. A larger battery could be put in the pocket of the experimental participant, but the power cable will pull on the device, introducing motion artifacts.

4. Support for the transducer selected. As explained in section 3.1, specialised circuitry is necessary to work with piezoelectric elements to avoid high passing the signal.

5. (**Preferably**) Support for an inertial measurement unit (IMU). Collecting inertial data alongside the audio data allows for multi-modal sensor fusion to be investigated.

It was difficult to find an off-the-shelf solution to the fourth requirement. While there exist expansion boards for Raspberry Pis and microcontroller boards to enable audio recording, they are not designed for contact microphones. They were also too large to be integrated into a wearable platform and do not include IMUs. To meet these requirements, some printed circuit boards were manufactured.

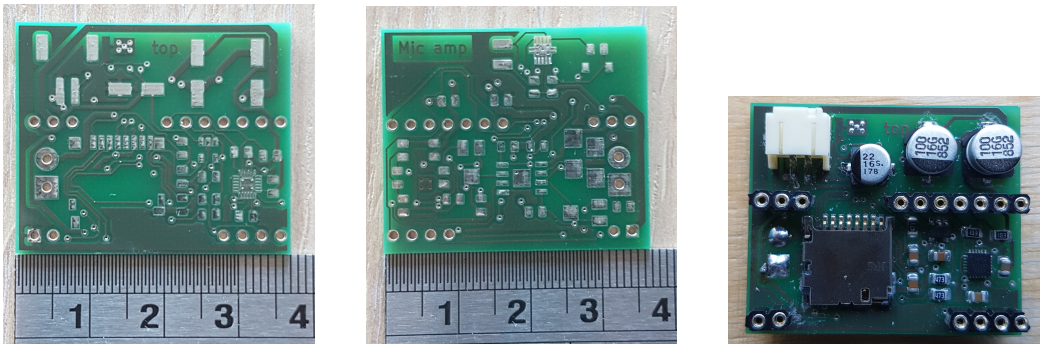
3.3 Hardware

This section gives an overview of the wearable hardware; design choices were made to optimise for datalogging.

3.3.1 Microcontroller

A Teensy 3.2 development board was selected to control the wearable. It was also possible to use a Raspberry Pi Zero, but the Teensy had several advantages over this option:

- The Teensy is Cortex-M4 based, and has lower power consumption than the Pi. The Teensy consumes 0.1W at full load with default settings [39]. The Pi uses 0.4W while *idle* with all unnecessary peripherals (e.g. HDMI) disabled [47].



(a) Top side of board. (b) Bottom side of board. (c) Top side when soldered.

Figure 3.3: Photos of the printed circuit board.

- Audio datalogging projects have been built on top of this platform [63]. The board manufacturer has written a dedicated audio library with a code generation tool [58].
- The Teensy has a high speed 16-bit ADC onboard suited for audio capture. The Pi does not have an ADC.

Although the Teensy is slower than the Pi, it is fast enough to perform basic real-time signal processing if it was found to be necessary.

3.3.2 Printed Circuit Board Implementation

The PCB was designed so that the Teensy could plug into it. The design had 2 layers, and had dimensions of 31x37mm. Most board area is devoted to audio, but some ancillary functionality was included.

Power

A boost converter was included, meaning the board could be powered from a LiPo battery. Additional regulation circuitry was included as the Teensy's onboard regulator was insufficient to power a microSD card.

Approximately 25% of board area was devoted to keeping the analogue power supply as clean as possible, as this impacts the audio quality. Large bulk capacitors suited to dampening the frequencies of interest were included, along with extensive use of decoupling capacitors on ICs (more aggressive than normal practice). The digital and analogue grounds were kept separate.

Inertial Sensing

An IMU was integrated into the board to allow for the simultaneous collection of inertial data. The InvenSense MPU-9250 [41] was chosen for the following reasons:

- It has low power consumption with low noise on readings.
- Several libraries for interfacing with it exist due to its popularity.

To avoid contention on the SPI bus, which was used to communicate with the microSD, the IMU was attached to the I2C bus.

Audio

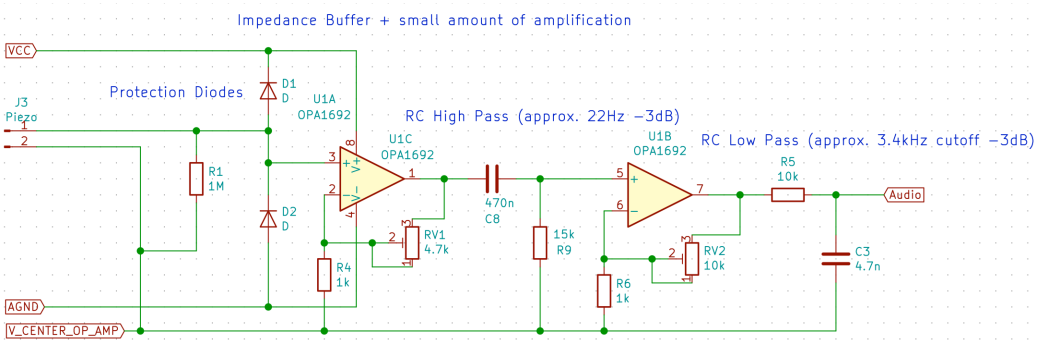


Figure 3.4: Schematic of the amplification circuitry used.

Connecting the microphone directly to the ADC yields poor results as there would be a poor impedance match which would high-pass the signal. The microphone output would also not use the full ADC range, resulting in lower resolution record-

ings than would otherwise be obtainable. It is necessary to amplify the signal while preserving the low frequency components.

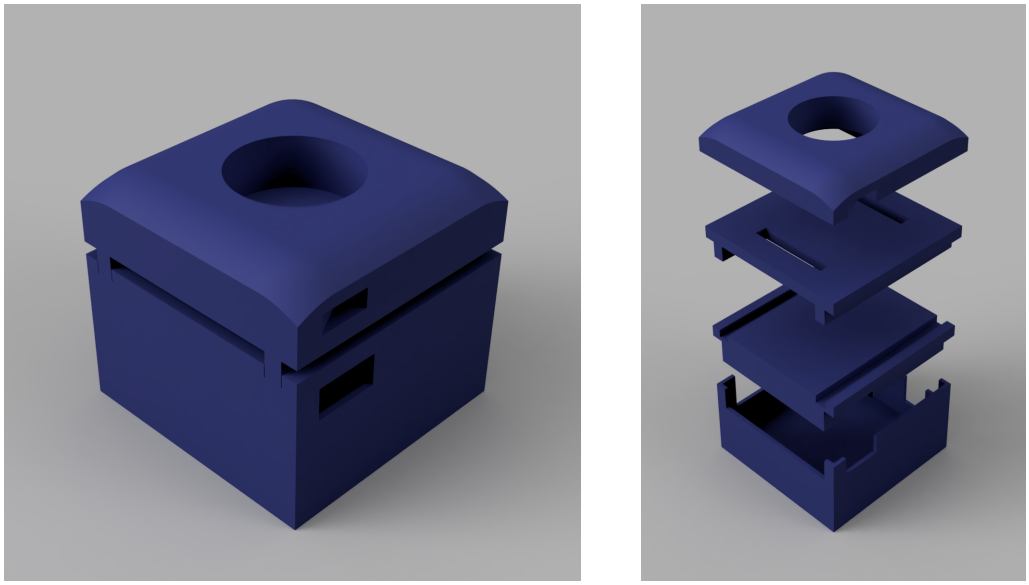
A buffer circuit was built using a single operation amplifier stage to improve the impedance match. Piezo voltage spikes were handled using transient voltage spike diodes. The buffered signal was fed into passive RC high-pass and low-pass filter stages and amplified using non-inverting amplifiers. The amplified signal was then connected to the Teensy's ADC. The schematic can be seen in fig. 3.4. The low-pass cut-off is approximately 3.4kHz, below the microphone's resonant frequency, but above the frequencies of interest.

The final PCB design used a low-power audio operational amplifier [45] designed for wireless microphones. An alternative design using a charge amplifier in place of the buffer circuit was also evaluated while prototyping. As the microphone wire was short and shielded, this design offered little benefit over a voltage amplifier in practice. Another consideration was that audio operational amplifiers are optimised for voltage amplification.

3.3.3 Casing

An enclosure for the PCB and microphone was built and mounted onto an elastic strap that could be worn around the torso. The wearable resembles commercially available chest mounted heart rate monitors [46, 14]. The enclosure was designed using Autodesk Fusion 360 [9] and 3D printed.

An initial iteration was unsuitable as the band was attached to the bottom of the casing, inducing a moment, and causing motion artifacts to be amplified. A second iteration was built which accounted for these issues by splitting the wearable into two halves. One half contained just the microphone enclosure; it was designed to be as thin as possible. The other half contained the electronics. The battery was moved into a separate enclosure on the band to minimise the weight localised in a single area. The parts were printed with less "infill" where possible, to reduce weight; the reduced structural rigidity was not an issue. Renders of the final design are shown in fig. 3.5.



(a) Constructed (b) Exploded view

Figure 3.5: Renders of the final casing used for the wearable.

The final wearable is shown in fig. 3.6. A button and LED was added to allow participants to start the recording and monitor the wearable status. This wearable had a depth of 37mm, which is shallow enough to fit under loose clothing. Future iterations could reduce the depth with tighter integration of the microcontroller module.

3.4 Software

A datalogging program was written for the Teensy using C++. The program continuously collected audio and IMU data and wrote it to a microSD card. The datalogger used several open-source libraries[2] to achieve this functionality.

It is not possible to simultaneously maintain two files on the microSD card, as access is slow when switching between files which causes samples to be dropped from RAM. Instead, data is dequeued from buffers for each sensor, and placed

[2] all licensed under MIT licenses

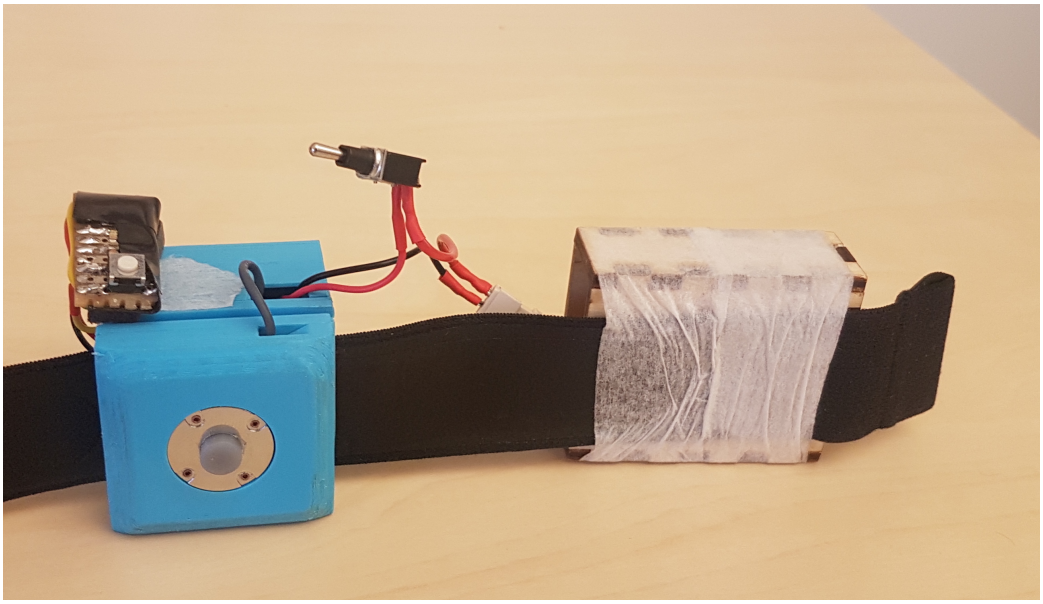


Figure 3.6: The final casing, as used during the data collection.

into a 512 byte block with a fixed memory layout. A Python script was written to interpret the blocks.

3.5 Device Evaluation

The wearable was assessed to ensure that it could be used for data collection. It was checked that the recordings made by the device had no dropped samples, and that the IMU and audio data were synced.

It was possible for the device to run for 3 hours off a 560mAh battery while continuously logging to a microSD. This was acceptable for data collection as individual trials would not take this long.

3.5.1 Signal Defects

It was observed that there was noise present on the audio signal at specific frequencies. Four fundamental frequencies were observed:

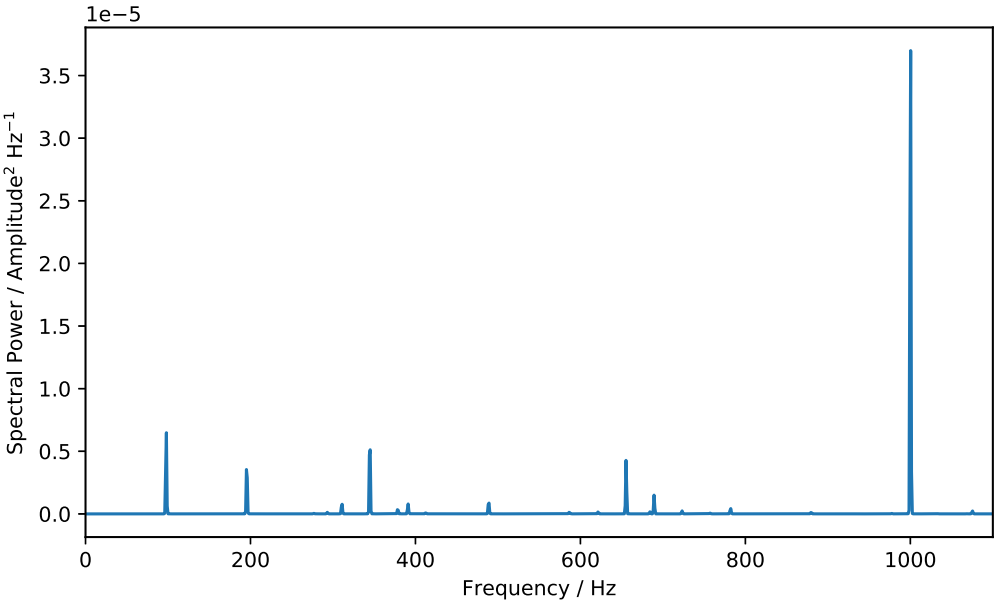
- 100Hz — corresponding to ground-loop buzz.
- 343 and 654Hz — related to microSD communication.
- 1000Hz — corresponding to the IMU sampling frequency.

These freqencies can be seen in fig. 3.7a. The presence of these defects indicates that the board design has power supply defects. The latter two defects are the result of digital noise reaching the analogue circuitry. As explained in section 3.3.2, significant effort was devoted to keeping the supply clean. There are two sources of noise:

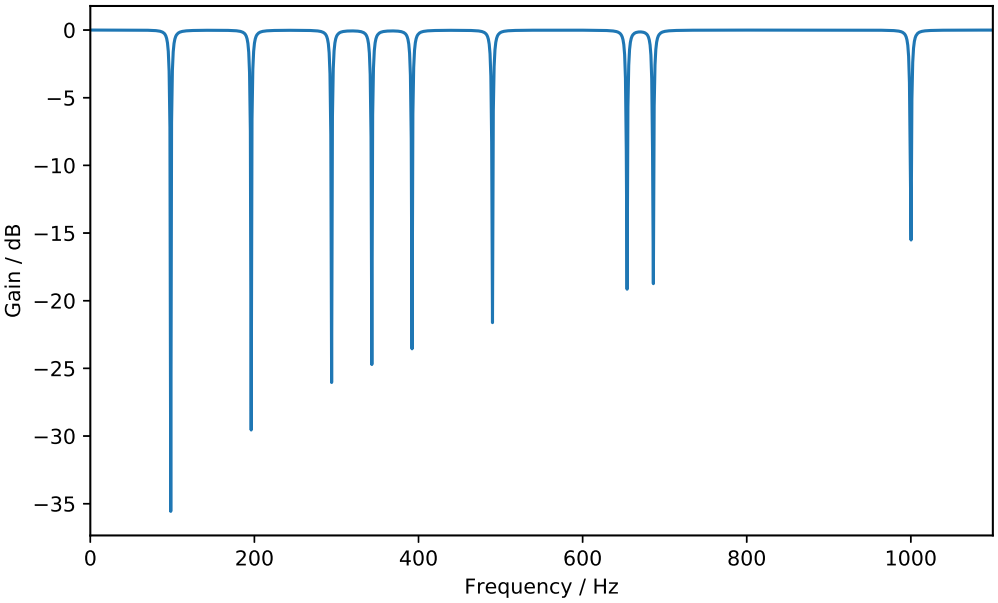
1. Insufficiently smooth positive rail.
2. Poor grounding causing noise on the digital ground to leak to the analogue ground.

Poor grounding is probably the primary source, and it explains the presence of 100Hz buzzing. Some fluctuations in the positive rail will usually be filtered by the operational amplifier IC's power circuitry; this filtering is not commonly done at the ground rail. However, it is impossible to be certain of this conclusion without taping-out new designs. It is difficult to know ahead of time whether the design sufficiently eliminates noise: several revisions are usually required. Only one iteration was built during this project, as these defects are trivial to remove due to their extremely narrow spectral width. Figure 3.7 has power spectral density plots before and after a comb filter was applied using forward-backward filtering (MATLAB `filtfilt`).

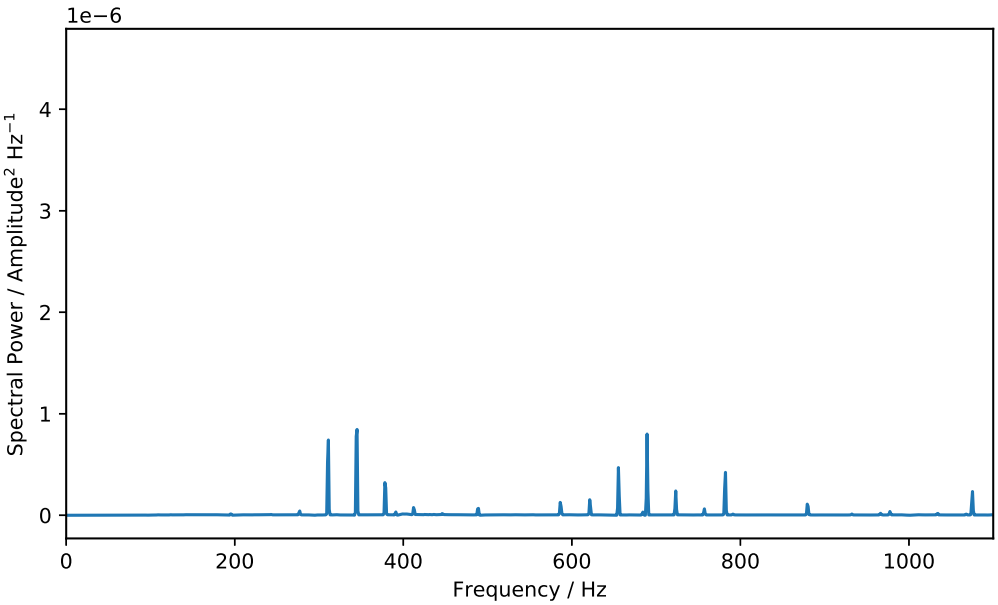
As this project is continuing after submission, it is likely that a new board revision will be made. For future revisions it would be sensible to incorporate a separate linear-dropout voltage regulator for analogue circuitry. Another improvement would be adding extra layers to the board to improve grounding; this was not done in the original design as moving to 4 layers would quadruple the fabrication cost.



(a) Power spectral density for raw recording made by the wearable.



(b) Frequency response of filter applied to raw recordings.



(c) Power spectral density after applying comb filter. *Note: different scaling for y-axis.*

Figure 3.7: Filtering applied to clean up the raw recording made by the wearable. Power spectral densities calculated using Welch's method.

Chapter 4

User Study

Before conducting the user study, the optimal placement for the device was investigated. The experimental protocol, along with motivation for the design, is provided. Finally, qualitative conclusions from the collected data are presented.

4.1 Optimal Placement for the Wearable

Various placements for the device were evaluated. As explained in section 2.2.1, the placement needs to be *discreet* otherwise users will not adopt the device. Four possible placements, which are often evaluated during diagnostic auscultation by physicians, were considered, marked in diagram fig. 4.1:

1. Bottom of sternum.
2. On the back, behind the heart.
3. Bottom of ribcage (offset towards with heart).
4. At the top of the chest (offset towards the heart). This placement was selected to be on top of the bronchus, which was hypothesised to allow better transmission of respiratory sounds.

Placement 1 is impractical for women and some men due to the necessity of

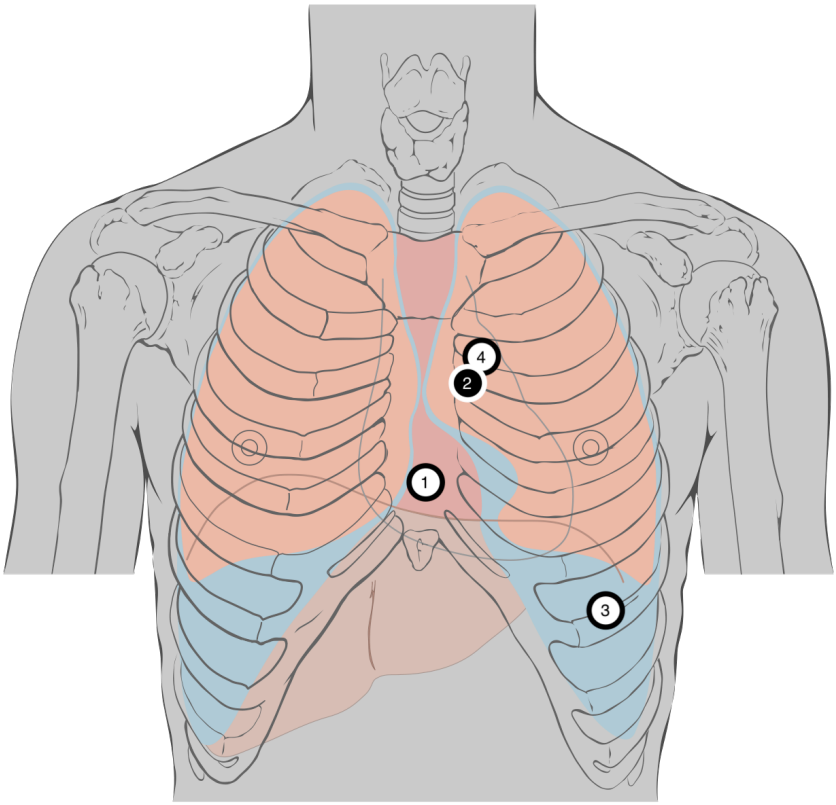


Figure 4.1: Placements considered for the wearable. Note that placements 1, 3 and 4 are on the chest, while 2 is on the back. Source image: [37]

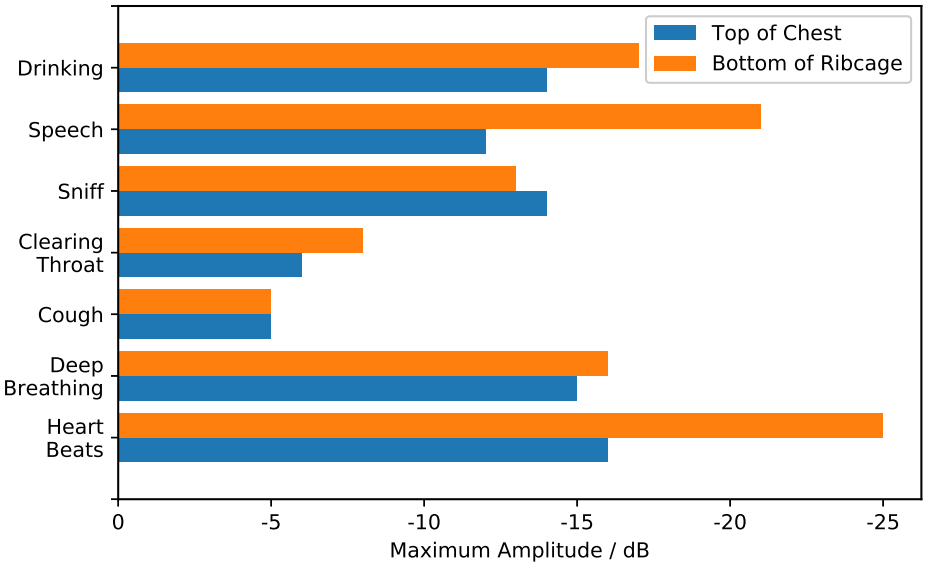


Figure 4.2: The maximum amplitude in the audio for different events of interest for the two placements evaluated (3 and 4 in fig. 4.1).

a strap to hold the device in contact with the skin. Placements on the sternum were generally unstable due to the surrounding curvature (which has a wide inter-person variance). Placement 2 is also problematic: shoulder blade movement displaced the device from the skin. Another downside of this location would be that the IMU data collected would not be usable for studying the seismocardiogram or respiration waveform.

Placement 3 and 4 were evaluated on the author, and the results are summarised in fig. 4.2. The top of the chest is the better placement: although respiratory events were comparable in amplitude across the placements, all other types of events were louder when using the placement at the top of the chest. It is worth mentioning, however, that for medical applications there are reasons to listen to the lungs. The crackles described in section 2.3 are lung sounds, while wheezes are tracheal.

4.2 Data Collection

Ethical approval was obtained to conduct a user study on healthy subjects. Experiment participants would wear the constructed device and would perform several activities under a variety of noise regimes.

The following data was recorded from users using the constructed device:

- Acoustic data from the microphone.
- All 3 axes of accelerometer data.
- All 3 axes of gyroscope data.

A ground truth device, the Zephyr Bioharness 3 [73], was also worn by participants. This device also used a band, but was worn around the bottom of the ribcage; fig. 4.3 indicates how the two devices are worn together. The ground truth device could record: (2 lead) ECG data, 3 axes of acceleration data, and the respiration signal. All this data was collected to validate the microphone. The ECG data can be used to assess the heart rate estimation accuracy. The respiration data is useful for assessing whether it is possible to monitor the breathing sounds and therefore derive the respiration rate from sound alone. Acceleration data was used to synchronise recordings between the two devices, and can be used for validating the presence of motion.

4.2.1 Procedure

An experimental procedure was designed in order to assess the microphone device's performance in a wide variety of scenarios. Three different noise regimes were defined:

1. Silence.
2. Music placed 1 metre from the experimental participant; average loudness at participant approx. 63dB.

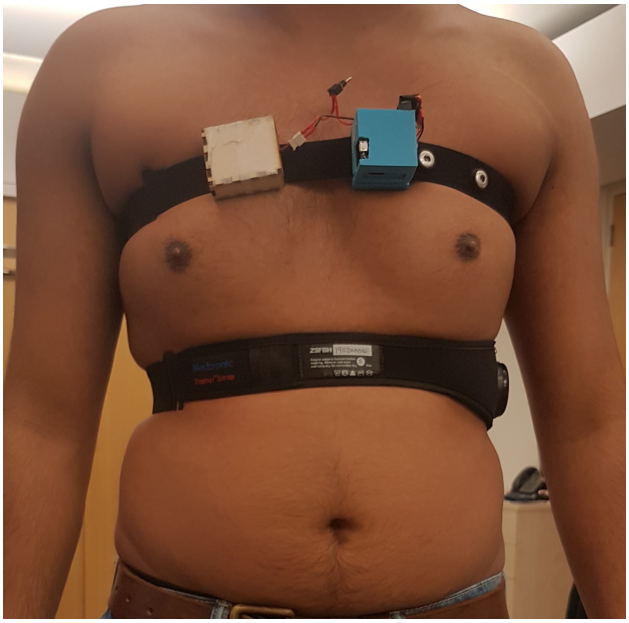


Figure 4.3: Photo of the author wearing the microphone and ground truth devices.

3. Background noise from a coffee shop placed 1 metre from the experimental participant; average loudness at participant approx. 42dB.

Participants were asked to perform the following activities:

1. Breathing normally (all regimes).
2. Breathing deeply (all regimes).
3. Coughing 10 times (all regimes).
4. Clearing throat 10 times (all regimes).
5. Swallowing 5 times (all regimes).
6. Drinking water (silent regime only).
7. Sniffing 10 times (all regimes).
8. Reading a new article (silent regime only).
9. Walking for 5 minutes (silent regime only).
10. Jogging for 5 minutes (silent regime only).

To the best of the author's knowledge, this is the most extensive evaluation dataset of a chest-mounted audio-based device: several noise regimes are assessed, along with several activities that are of interest. In particular, this is the

first dataset which adequately considers *motion* and its impact on the recordings obtained from the body. There is sufficient data to support future analyses beyond the scope of this dissertation: examples include respiration rate or respiratory event (e.g. coughing and sniffing) monitoring. An extensive analysis for either of these problems would constitute a publication at a top-tier conference. Ethical approval was obtained to release this dataset to the wider research community.

The procedure attempts to distinguish between digestive sounds (swallowing, drinking) and ventilatory sounds. This is of interest as there is little acoustic data for tracking these digestive sounds, and it is unclear if a chest mounted device could be used to adequately discern them. Motion activities are not performed on a treadmill, as in many studies, because it known that walking on a treadmill does not correspond exactly to forces experienced by the body when walking on ground [68].

As it is intended that this dataset will be released, there is a risk of de-anonymisation for experimental participants. Participants were warned that the dataset may be released, and the procedure was designed to minimise the risk of participants divulging personal information; one example is the procedure involving the reading of news article and no other speech. Additionally, only healthy individuals were recruited for the study to prevent identification from uncommon medical conditions.

9 participants were recruited (3 women). Each trial yielded approximately 30 minutes of data.

4.2.2 Issues Encountered During Collection

As would be expected for a user study of this kind, there were initial issues. The first problem was associated with power: the power connector selected had a tendency to become loose, causing the recording to cease. This was fixed by changing the connector.

The main issue observed during the collection was microSD card failures. Some

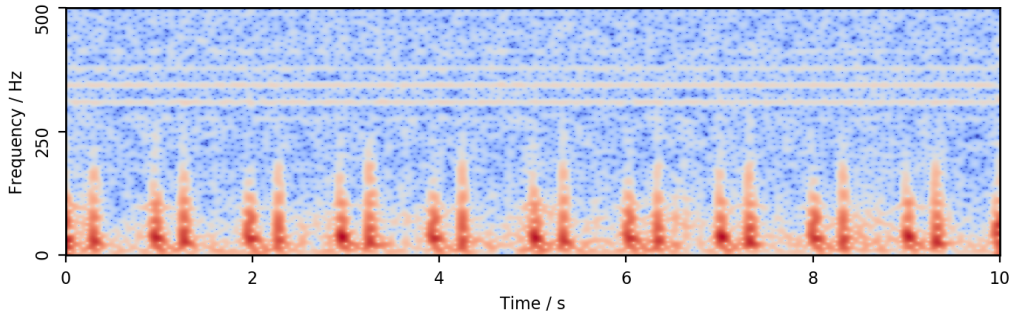
microSD cards could not sustain continuous recording for extended periods of time, and would stop non-deterministically; however, each card was fine after power cycling. This behaviour varied by card manufacturer. It would be prudent to use a more robust alternative such as eMMC storage in future device revisions. To minimise the risk of data loss, later participants made several short recordings rather than one extended recording.

4.3 Qualitative Findings from Data

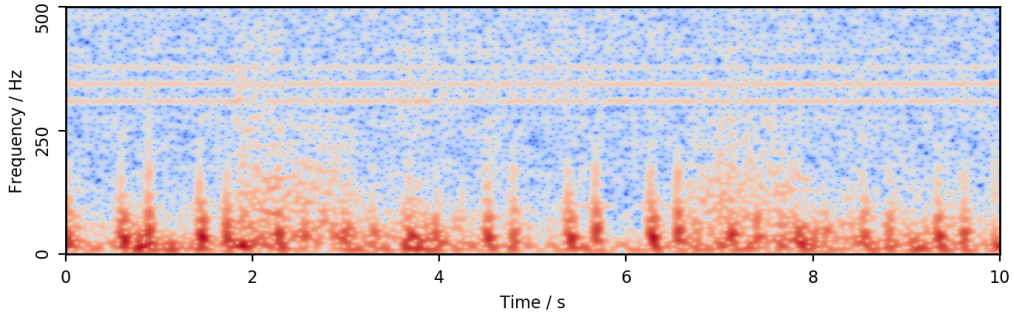
Figure 4.4 has several spectrograms for the different activities participants were asked to perform. Qualitatively it did not appear that the different noise regimes had any effect, so no distinction is made. This noise robustness validates the reasoning given in section 3.1 for choosing a contact microphone.

The heart sounds are clearly visible during periods when the participant was resting (fig. 4.4a). Normal breathing does not appear to be discernable for most participants; this sound has little power, and requires precise placement of the device to capture. The author evaluated this hypothesis and found that there was a region of approximately 1cm^2 where normal breathing sounds could be identified. Deep breathing (fig. 4.4b), however, is audible for every participant and can be seen in the spectrogram. Other respiratory events can also be identified: coughs, throat clearing, sniffing and speech are all visible in the spectrogram and are likely distinct enough that a convolutional neural network classifier could be accurately classify them from the spectrogram. Swallowing and drinking sounds are mostly contained below 70Hz, with the exception of gulps taken when drinking: these are loud, and can have significant energy up to 600Hz. This is interesting as it was difficult to find evidence of these sounds being audible during auscultation of the chest.

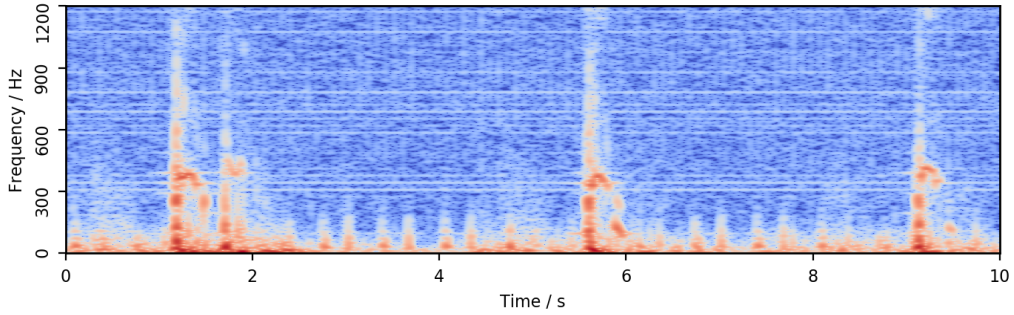
The spectrograms for walking and running demonstrate a stark difference: during walking the spectral energy is contained below 100Hz, which suggests that it may still be possible to discern individual heart sounds, as they have significant energy between 100 and 200Hz. Running, however, induces energy at frequencies up



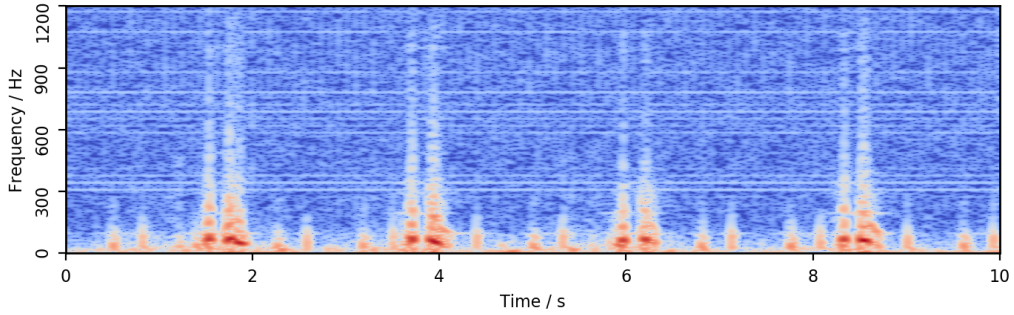
(a) Normal breathing



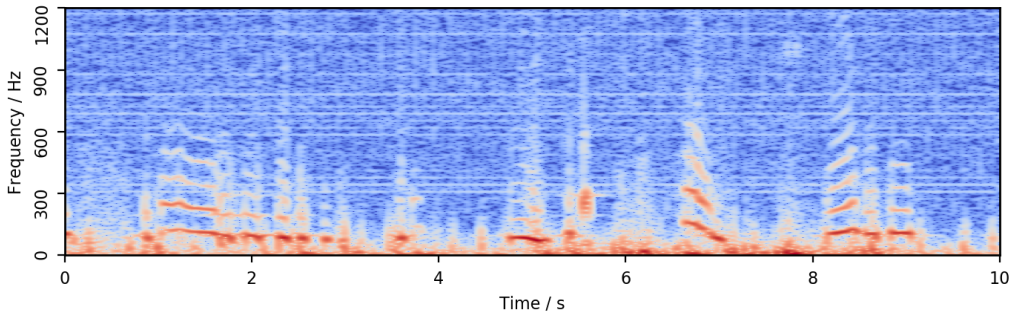
(b) Deep breathing



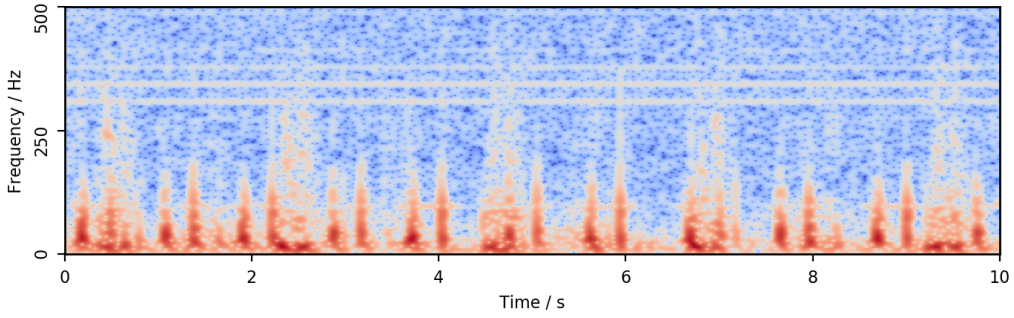
(c) Coughing



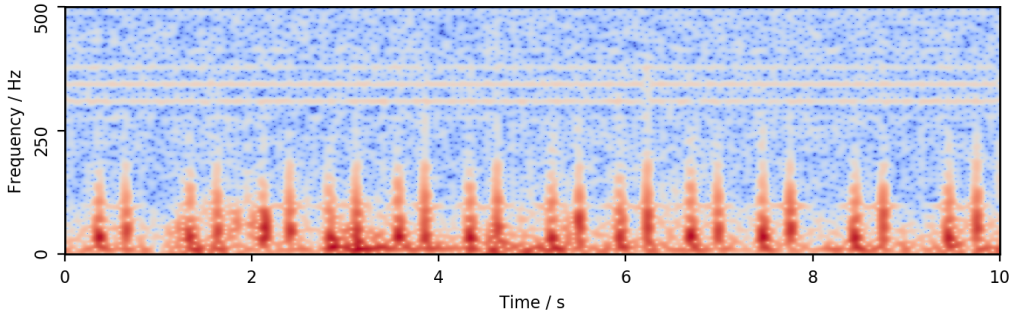
(d) Clearing throat



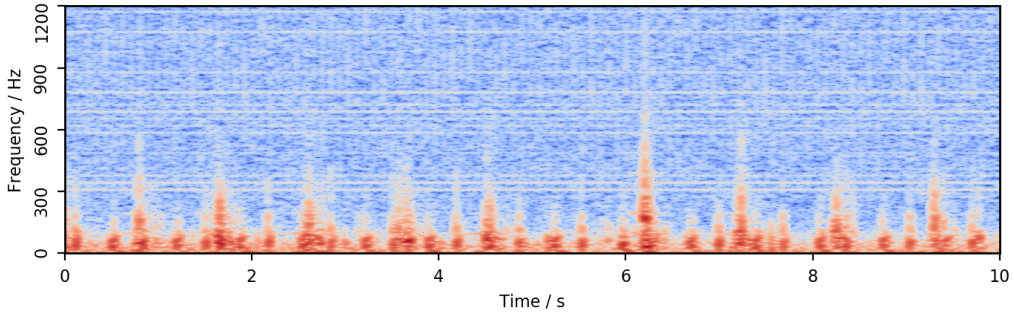
(e) Speech



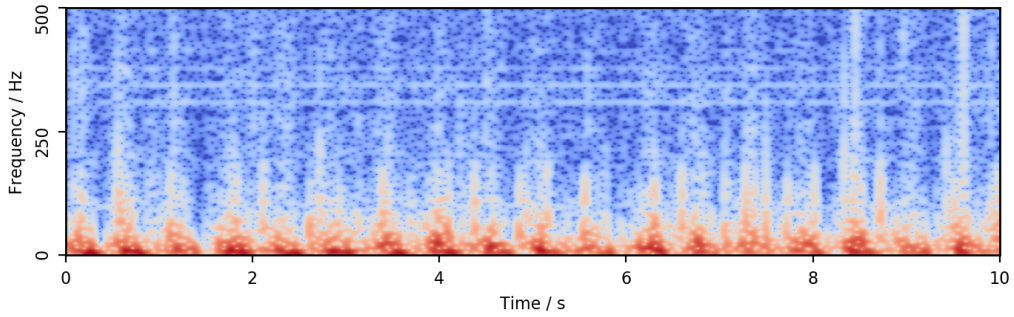
(f) Sniffing



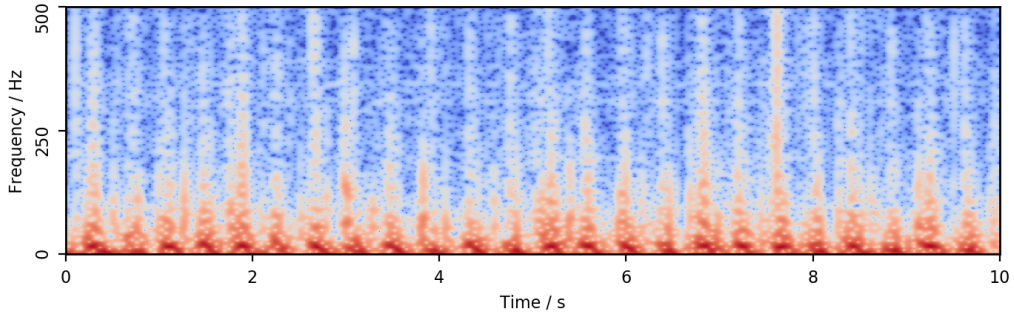
(g) Swallowing



(h) Drinking



(i) Walking



(j) Running

Figure 4.4: (Power) spectrograms for different activities evaluated during the user study.

to 300Hz in some individuals. This difference suggests that it may be possible to use audio from the body to do activity recognition, without using an IMU as most wearables do to perform this functionality. It is worth noting that it is possible to see sudden increases in energy at frequencies below 50Hz for both spectrograms: these are footsteps. This audio does not correspond to sound as we would normally perceive it: it is really a measure of body tissue vibration, and it extends into the infrasonic frequencies. It was hypothesised that these spectra observed during motion were due to clothing rubbing against the device during motion; the author evaluated this hypothesis by using the device while topless and observed no difference.

Chapter 5

Continuous Heart Monitoring

In this chapter, two different algorithms for continuous heart monitoring are described. The first algorithm, which estimates the heart rate, is designed to be cheap to compute continuously, to the detriment of robustness to activities such as speech or motion. It is argued that these events are rare: the median number of steps walked per day by adults in the UK was under 2000 according to one study [7], which corresponds to approximately 8 minutes per day. A similar argument can be made for other events, such as speech. Another algorithm, which segments the audio into the phases of the cardiac cycle, is also proposed; this algorithm is more computationally expensive, but is more noise robust and yields medically relevant information that a heart rate estimate does not.

The algorithms described in this section are adapted from work in the PCG analysis literature. Researchers in this field do not analyse their work in the context of limited computational resources or robustness to noisy measurements. For wearable devices, however, this is a paramount consideration: several novel adaptations to the original algorithms are described to enable usage in the wild.

5.1 Heart Rate Estimation

Springer et al. [55] provided a survey on various techniques that have been proposed to estimate the heart rate robustly from noisy PCGs. Each technique used the following structure:

1. Some kind of preprocessing, usually including wavelet denoising.
2. Extraction of signal envelope, over the *entire* recording. Empirically, the authors found that the Hilbert envelope was the best choice.
3. Calculating the autocorrelation of the envelope, and choosing the most prominent peak which is within a plausible lag. Too short a lag usually corresponds to overlap between the S_1 and S_2 heart sounds; too large a lag does not correspond to the heart beat.

Springer et al. ran their algorithms over entire PCGs, which varied in length to over 120 seconds, but type of processing is implausible with low power embedded processors: there is insufficient RAM to fit the samples and the $\mathcal{O}(n \log n)$ runtime of the autocorrelation makes real-time processing infeasible. Wavelet denoising is also an expensive procedure.

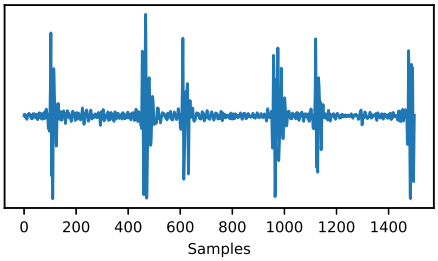
Algorithm 1 is proposed, which considers the criticisms of the original algorithms from the literature. One novel change is that the signal is divided into windows, and for each window a peak from the autocorrelation peak is chosen. This limits the RAM necessary for processing, and allows for real-time estimation. There is a downside to using smaller windows: they are less noise robust. To account for this, the lags are post-processed, and outliers are rejected by comparing to a window of previous measurements. In practice, outliers *over*-estimated the heart rate. Consider the autocorrelation spectrum when the wearer coughs halfway between two heart beats: in this case, the autocorrelation spectrum would have a peak corresponding to the distance between a beat and the cough, which could dominate the true peak. A simplistic scheme to detect outliers is used: any peaks above a fixed percentile of the previous window of measurements are rejected. The remaining lags are Kalman filtered [61] to arrive at a final heart rate estimate, as the technique can robustly handle missing measurements. Some applications

Algorithm 1 Efficient heart rate estimation. For simplicity the pseudocode assumes batch processing.

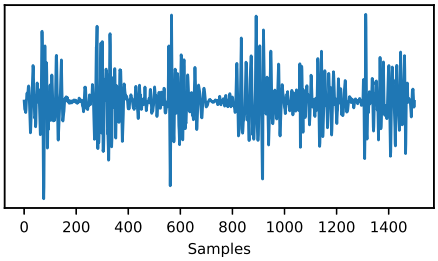
```
function HeartRateEstimate(x, minbpm, maxbpm, fs, windowSize,
percentile)
  // Filtering motivated by heart sound frequencies given in
section 3.1
  x = bandpass(25, 175)
  maxlag = (60 / maxbpm) × fs
  minlag = (60 / minbpm) × fs
  lags = list()
  for window in x do
    envelope = HilbertEnvelope(window)
    autocorrelated = Autocorrelate(envelope, maxlag)
    lags.append(ChoosePeak(autocorrelated, minlag, maxlag))
  bpm = 60 / (lags / fs)
  outliers = OutlierReject(bpm, windowSize, percentile)
  bpm = KalmanFilter(bpm, outliers)
  return bpm
```

use the Kalman filter's mean and variance estimates to reject outliers; in this case, however, this is ineffective as the distribution of observations is bi-modal, and cannot be approximated by a Gaussian. The final implementation used 2 second audio windows, and rejected any windows that were above the 70th percentile of the previous 7 windows. Using 2 second windows means that the samples can fit into a 1024-point FFT while still satisfying the Nyquist limit.

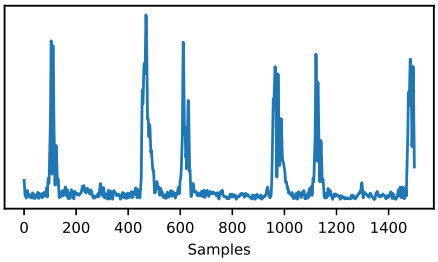
Visualisations of the stages of signal processing can be seen in fig. 5.1. A stark difference can be seen between the resting and walking activities: when the user is walking, the autocorrelation peaks correspond to footsteps. The most prominent peak, at a lag of approximately 250 samples, corresponds to 2 footsteps per second, corresponding to walking pace [62]. When the user is still it is possible to determine the S_1-S_2 distance (the peak at 150 samples), along with the most prominent peak when the two beats align best (the peak at 500 samples).



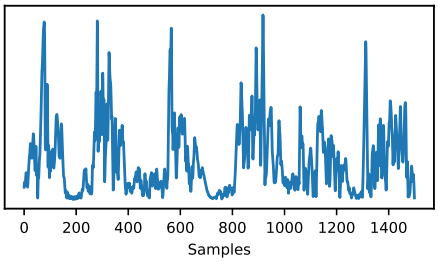
(a) Time domain (still)



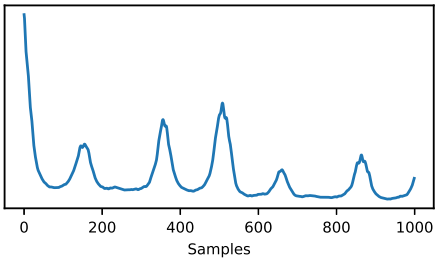
(b) Time domain (walking)



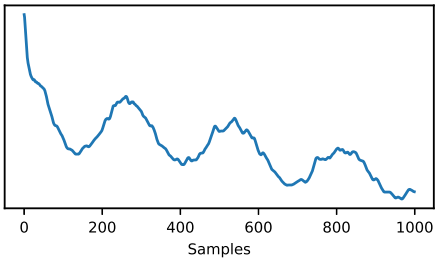
(c) Hilbert envelope (still)



(d) Hilbert envelope (walking)



(e) Autocorrelation (still)



(f) Autocorrelation (walking)

Figure 5.1: Visualisation of the stages in algorithm 1, for both still and walking situations. Audio sampled at 500Hz.

5.2 Segmentation

The following section describes an algorithm for segmenting the audio into phases in the cardiac cycle. By incorporating knowledge of the cardiac cycle the algorithm is more robust to noise. It can be used to estimate heart rate variability (HRV), which is the standard deviation of intervals between R-peaks in the ECG[1], and a useful proxy for measuring stress levels [60] (amongst other uses).

5.2.1 State of the Art

Springer et al. proposed an algorithm for PCG segmentation, which used a hidden *semi*-markov model (HSMM) to model transitions between states [54]. The audio is assumed to continously cycle through 4 states: S_1, systole, S_2 and diastole. An HSMM allows for the state residence time to be modelled, as the transition probabilities are dependent on the time that has elapsed since entry to the current state. An ordinary HMM models residence distributions as geometric, which is a poor approximation for this problem. Their work assumed that the duration of each phase was normally distributed.

The emission probabilities for each state were modelled using logistic regression trained from four features extracted from the PCG. Each feature was a type of envelope extracted from the signal. The envelopes were downsampled and labelled as belonging to one of the four phases using reference ECG data which had been collected alongside the audio.

5.2.2 Adaptations for Continuous Monitoring

A reproduction of this technique was attempted. Following Springer's procedure, the audio data was labelled using the ground truth ECG: it is possible to localise the start of the S_1 sound from the ECG R-Peak, and the start of the S_2 sound from the ECG T-Wave. When labelling, each S_1 and S_2 sound was assumed to

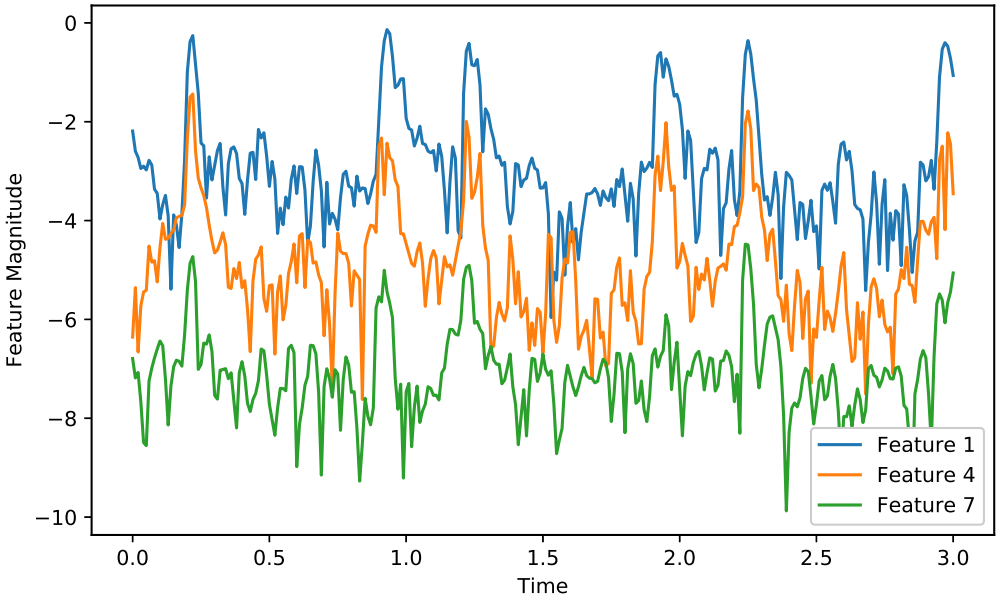
[1] the phases of an ECG can be seen in fig. 2.1b

last the mean duration reported by Schmidt et al. [52] for the two types of sound on their expertly annotated dataset. Unfortunately, results on the collected data were extremely poor when using the technique as described. The original features proposed are insufficiently robust to noise: envelope based features are disrupted by transient changes in signal amplitude.

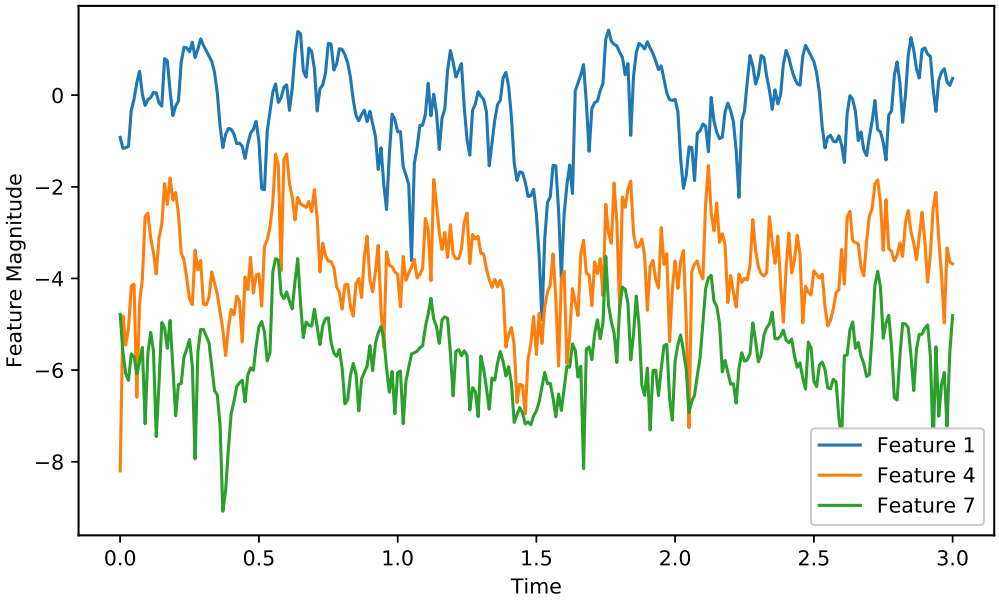
In section 4.3, it was observed that different activities introduced power into different parts of the spectrum. Motivated by this observation, the magnitudes of coefficients from the Short-Time Fourier Transform (STFT) were used as features instead. The reasoning is that a classifier could learn to ignore spikes in coefficients that correspond to known noise profiles—so long as the rest of the coefficients are consistent with a heart sound. For numerical robustness, the log-magnitude is used as a feature. An audio sampling rate of 500Hz was assumed, which allows for frequencies below 200Hz to be distinguished, assuming a high-quality anti-aliasing filter. The STFT was performed on 16 samples, using a hop length of 5 samples; the samples were windowed using the Hann function. These choice of coefficients yields 100×9 features per second.

A plot of the features against time is given in fig. 5.2. When the user is still, consistent spikes in all features can be observed, corresponding to heart sounds. Determining the location of the heart sounds from walking data is difficult, but not impossible. As explained, the noise introduced by walking primarily affects low frequencies (sub-100Hz), and hence a classifier could—in principle—learn to handle these trends. It is possible to learn the approximate magnitude for each feature that corresponds to a heart sound: if the magnitude is too high, then it is likely affected by noise.

Logistic regression was ineffective when noise was present, due to its ability to learn only linear decision boundaries. Random forests were substituted as an alternative which could learn more complicated relationships, while admitting efficient inference on a microcontroller. After making this substitution, performance improved to an acceptable level. The random forest was trained to predict the presence of either type of heart sound, reducing it to a binary classification problem. At time t it was assumed that the emission probability for the S_1 and S_2 states was the same; the same procedure was applied to the systole and diastole



(a) Still



(b) Walking

Figure 5.2: Plot of spectral features used for segmentation against time. Higher-indexed features correspond to higher frequencies. The features are extracted from the same audio samples used to create fig. 5.1.

states. This technique was highly effective: it is difficult for the random forest to accurately disambiguate between the states when trained in a one-versus-rest fashion. This is unsurprising as there is little to disambiguate the systole and diastole states, which correspond to a *lack* of heart sounds. The S_1 and S_2 sounds are also highly similar, differing mainly in duration. By tying the emission probabilities, the inference of the most likely path through the states relies on duration information. This is relatively easy as the systole and diastole states have notably different durations: values of 0.128 ± 0.062 and 0.356 ± 0.121 seconds for the two states were observed in the collected dataset.

To prevent overfitting, and minimise inference time, a forest of 10 trees with a maximum depth of 8 was used. Increasing these hyperparameters yielded marginal improvements in classification, with correspondingly longer inference time. Another aspect that was explored was the use of Mel-frequency cepstrum coefficients (MFCCs), which are derived from the STFT coefficients, and are a popular choice for features when working with audio. MFCCs offered no benefit over raw STFT coefficients in this case, and were not used as they require extra computation to derive.

Estimating Heart Rate and Variability

The labelling generated by algorithm 2 must be post-processed to obtain the heart rate and variability. Algorithm 3 indicates the algorithms used to obtain these values. The intervals between heart beats are found from the labels; intervals are the differences between the *start* of two S_1 states.

It is not necessary to filter outliers before Kalman filtering when estimating the heart rate; the HSMM assumptions prevent a bimodal distribution in inter-beat intervals. However, when calculating the heart rate variability, a standard technique [26] from the biomedical signal analysis literature is used. If the inter-beat time deviates from the mean of the previous 4 inter-beat times by more than 30%, it is assumed to be an outlier.

Algorithm 2 Heart rate segmentation.

```
Require: N: Number of states;
  T: Length of sequence to decode;
  B: Emission probabilities;
  A: Transition matrix;
  π: Initial state probabilities;
  p: Duration matrix; p_j(d) is the probability of staying in state
  j for duration d;
  d_max: Length of duration matrix
  function HsmmDecode(N, T, B, A, π, p, d_max)
    Initialise δ, ψ and D matrices (shape T × N)
    for j in 0...N − 1 do
      δ_0(j) = π_j
    for t in 1...T − 1 do
      for i, j in (0...N − 1, 0...N − 1) do
        Let e(d) = max_{i≠j}{δ_{max(0,t−d)}(i)·A(i,j)}·p_j(d)·∏_{k=max(0,t−d)}^{t} B(j,k)
        // This is similar to Viterbi for a normal HMM
        // NOTE: This is an inner loop
        δ_t(j) = max_{1≤d≤d_max} e
        D_t(j) = arg max_{1≤d≤d_max} e

        ψ_t(j) = arg max_{0≤k≤N−1}{δ_{t−D_t(j)}(k) · A(k,j)}

    // Decode from calculated matrices
    t = T − 1
    q*_t = arg max_i δ_{T−1}(i)
    while t > 0 do
      d* = D_t(q*_t)
      q*_{t−d*→t−1} = q*_t
      q*_{t−d*−1} = ψ_t(q*_t)
      t -= d*
    return q*

  function SegmentHeartCycle(x, classifier)
    // log-magnitudes of STFT coefficients
    features = ExtractFeatures(x)
    // Estimate emission probabilities used during decoding
    probs = estimateEmissionProbs(classifier, features)
    return HsmmDecode(4, Len(probs), probs, ...)
```

Algorithm 3 Estimation of heart rate and variability from labels generated by the segmentation algorithm. Assumes batch processing for simplicity.

```
function EstimateHeartRate(labels)
    deltas = FindTimesBetweenBeats(labels)
    bpm = 60 / (deltas / labelSampleRate)
    bpm = KalmanFilter(bpm)
    return bpm


function EstimateVariability(labels)
    deltas = FindTimesBetweenBeats(labels)
    retainedDeltas = List()
    for i in 1...Len(deltas) - 1 do
        // Find the mean of the 4 previous inter-beat times
        window = deltas[Max(i - 4, 0): i]
        m = Mean(window)
        // Reject if more than 30% away from the local mean
        if m × 0.7 ≤ deltas[i] ≤ m × 1.3 then
            retainedDeltas.Append(deltas[i])
    return StandardDeviation(retainedDeltas)
```

5.3 Evaluation

The algorithms will be evaluated from two perspectives: accuracy and their viability of implementation on plausible hardware.

5.3.1 Accuracy

Both algorithms will have their heart rate estimations evaluated with reference to the ECG. The quality of the segmentation and heart rate variability estimates will also be assessed for the segmentation algorithm.

Since the segmentation algorithm relies training a random forest, its results are reported using leave-one-person-out cross-validation. The experiments were repeated 10 times as the classifiers used to calculate emission probabilities have random initial state.

ECG Preprocessing

Raw ECG data was processed using the BioSPPY [67] and Neurokit [1] libraries.

There is no agreed algorithm for estimating the heart rate from ECG data. Using the raw signal derived from the time between beats yields noisy results because of heart rate variability. It was decided to apply exponential smoothing to the raw signal i.e. $s(t) = \alpha x(t) + (1-\alpha)s(t-1)$. $\alpha = 0.075$ was chosen as a compromise between rejecting high-frequency noise while still preserving local trends.

For heart rate variability estimation, the same outlier rejection method as used in algorithm 3 is used [26].

Rate Estimation

The rate estimation algorithms were evaluated over every activity-noise regime pair. The estimates were compared to the ground truth every 2 seconds.

Estimation results for the autocorrelation- and segmentation-based algorithms are given in table 5.1 and table 5.2 respectively. A visualisation of the tracking for each algorithm over an entire data collection trial is also shown in fig. 5.3.

There are several points for discussion:

- Accuracy when resting is competitive with commercially available chest-mounted heart rate trackers; one study indicated a popular device had a mean percentage error of 0.8% [17], but it should be emphasised that the authors do not explain their method for calculating the ground truth. With the segmentation method, a mean percentage error of approximately 3.34% was obtained in the worst case, but the median percentage error was below 0.33% for each scenario. The first section of fig. 5.3b demonstrates this visually: the estimate tracks the ground truth almost perfectly. The performance with the autocorrelation-based algorithm is accurate to within 2% median percentage error even under noisy conditions, which is sufficiently accurate to be useful.

Activity	Noise Regime	Median Absolute Error / BPM	Mean Absolute Error / BPM	Median Percentage Error	Mean Percentage Error
Rest	Silence	2.47	6.73	3.24	10.95
Rest	Music	1.27	3.25	1.86	4.58
Rest	Conversation	1.22	1.65	1.72	2.17
Deep Breathing	Silence	2.41	2.98	2.98	3.64
Deep Breathing	Music	2.02	2.69	2.36	3.08
Deep Breathing	Conversation	1.79	2.06	2.11	2.42
Coughing	Silence	5.05	7.69	5.69	8.07
Coughing	Music	10.33	12.17	13.18	15.05
Coughing	Conversation	11.07	12.35	13.37	14.63
Clearing Throat	Silence	5.68	9.66	8.43	13.10
Clearing Throat	Music	11.77	12.81	14.40	18.06
Clearing Throat	Conversation	9.68	12.85	12.85	17.62
Swallowing	Silence	4.03	7.04	4.82	9.05
Swallowing	Music	7.49	9.75	9.63	13.27
Swallowing	Conversation	5.04	11.39	6.44	15.26
Drinking	Silence	5.43	11.12	6.57	14.71
Sniffing	Silence	12.69	12.32	17.53	16.78
Sniffing	Music	5.06	7.99	6.73	10.71
Sniffing	Conversation	2.93	5.90	4.15	7.98
Speech	Silence	28.75	29.46	39.84	43.50
Walking	Silence	13.31	13.80	16.33	16.78
Running	Silence	25.61	23.73	22.61	20.46

Table 5.1: Accuracy of autocorrelation-based estimation algorithm (algorithm 1).

- It does not appear that the noise regime affects the accuracy of the estimation algorithms — or, it is necessary to collect a larger dataset to arrive at a definitive conclusion. For each activity, it is unpredictable which noise regime will yield the worst results: on multiple occasions the silent noise regime is worst.

  The results for sniffing and coughing under the silent regime appear to be anomalous and related to the way of the way the data collection was run; most experiment participants made multiple recordings, and these activities were the first in each recording. As the Kalman filter needs several samples to converge to an accurate estimate these results are worse than would be expected.

Activity	Noise Regime	Median Absolute Error / BPM	Mean Absolute Error / BPM	Median Percentage Error	Mean Percentage Error
Rest	Silence	0.23 ± 0.01	1.18 ± 0.08	0.33 ± 0.02	1.49 ± 0.09
Rest	Music	0.20 ± 0.01	2.91 ± 0.11	0.26 ± 0.02	3.34 ± 0.13
Rest	Conversation	0.14 ± 0.04	2.54 ± 0.12	0.22 ± 0.07	2.77 ± 0.13
Deep Breathing	Silence	5.18 ± 0.27	7.15 ± 0.11	6.04 ± 0.32	7.80 ± 0.12
Deep Breathing	Music	11.84 ± 0.43	14.20 ± 0.21	13.91 ± 0.56	15.07 ± 0.24
Deep Breathing	Conversation	8.47 ± 0.74	12.29 ± 0.30	9.94 ± 0.82	13.25 ± 0.34
Coughing	Silence	13.75 ± 1.45	13.61 ± 0.61	14.24 ± 1.38	14.33 ± 0.67
Coughing	Music	5.81 ± 0.40	8.59 ± 0.23	6.98 ± 0.49	9.53 ± 0.29
Coughing	Conversation	5.32 ± 0.32	8.73 ± 0.19	6.75 ± 0.57	9.57 ± 0.23
Clearing Throat	Silence	5.40 ± 0.57	7.90 ± 0.35	6.71 ± 0.85	8.96 ± 0.41
Clearing Throat	Music	4.14 ± 0.50	7.27 ± 0.33	5.70 ± 0.67	8.68 ± 0.49
Clearing Throat	Conversation	1.40 ± 0.18	4.78 ± 0.36	1.91 ± 0.19	5.31 ± 0.40
Swallowing	Silence	2.51 ± 0.78	5.65 ± 0.27	2.93 ± 0.90	6.28 ± 0.32
Swallowing	Music	3.24 ± 0.86	5.42 ± 0.31	3.98 ± 0.81	6.19 ± 0.35
Swallowing	Conversation	3.09 ± 0.53	8.11 ± 0.18	4.33 ± 0.76	9.46 ± 0.24
Drinking	Silence	7.07 ± 0.27	7.97 ± 0.30	8.23 ± 0.38	9.18 ± 0.34
Sniffing	Silence	2.89 ± 0.41	4.63 ± 0.29	3.63 ± 0.34	5.73 ± 0.44
Sniffing	Music	1.02 ± 0.22	3.85 ± 0.39	1.45 ± 0.35	4.79 ± 0.55
Sniffing	Conversation	1.18 ± 0.10	3.77 ± 0.22	1.64 ± 0.11	4.31 ± 0.26
Speech	Silence	7.55 ± 0.41	10.81 ± 0.26	11.11 ± 0.56	13.85 ± 0.37
Walking	Silence	6.11 ± 0.10	8.56 ± 0.09	7.35 ± 0.17	9.29 ± 0.11
Running	Silence	37.61 ± 0.40	38.99 ± 0.25	31.62 ± 0.31	30.64 ± 0.20

Table 5.2: Accuracy of segmentation-based estimation algorithm.

- The segmentation algorithm is usually more accurate, with the exception of the deep breathing activity. That activity's error is significantly larger than for any other type of respiratory event, including violent ones such as coughing or throat clearing. These results are likely related to a medical phenomenon mentioned in section 2.3.1: during inhalation, the second heart sound splits. As breathing sounds occupy the same portion of the spectrum as the heart sounds, it is difficult to accurately localise the second heart sound, causing segmentation performance to degrade.

- It is possible to obtain heart rate estimates during walking, confirming the hypothesis in section 4.3. A median percentage error of 7.35% was obtained using the segmentation algorithm. The autocorrelation-based algorithm is ineffective when walking: it appears to return the footstep

rate. As predicted in section 4.3, it does not appear to be possible to obtain any estimate when running.

- It is difficult to extract the heart rate during speech. It was hypothesised that better results would be obtained with women, due to the fundamental vocal frequency overlapping less with the heart sounds, but this was not observed. On inspecting fig. 4.4e it can be seen that during speech breath sounds can be identified, which is a possible explanation for why poor performance was observed for both genders.

Segmentation Accuracy

Activity	Noise Regime	Precision	Recall	F1
Rest	Silence	0.976 ± 0.001	0.960 ± 0.003	0.968 ± 0.002
Rest	Music	0.937 ± 0.004	0.893 ± 0.004	0.915 ± 0.004
Rest	Conversation	0.963 ± 0.004	0.934 ± 0.006	0.948 ± 0.005
Deep Breathing	Silence	0.864 ± 0.004	0.772 ± 0.004	0.815 ± 0.004
Deep Breathing	Music	0.776 ± 0.007	0.637 ± 0.008	0.699 ± 0.007
Deep Breathing	Conversation	0.819 ± 0.004	0.703 ± 0.005	0.757 ± 0.005
Coughing	Silence	0.693 ± 0.011	0.568 ± 0.010	0.624 ± 0.011
Coughing	Music	0.768 ± 0.009	0.682 ± 0.011	0.722 ± 0.010
Coughing	Conversation	0.749 ± 0.011	0.655 ± 0.011	0.699 ± 0.011
Clearing Throat	Silence	0.725 ± 0.012	0.658 ± 0.011	0.690 ± 0.011
Clearing Throat	Music	0.820 ± 0.013	0.765 ± 0.012	0.791 ± 0.013
Clearing Throat	Conversation	0.783 ± 0.011	0.736 ± 0.009	0.759 ± 0.010
Swallowing	Silence	0.902 ± 0.015	0.865 ± 0.021	0.883 ± 0.018
Swallowing	Music	0.866 ± 0.006	0.790 ± 0.010	0.827 ± 0.008
Swallowing	Conversation	0.829 ± 0.013	0.752 ± 0.013	0.789 ± 0.013
Drinking	Silence	0.801 ± 0.007	0.717 ± 0.005	0.756 ± 0.006
Sniffing	Silence	0.832 ± 0.005	0.797 ± 0.006	0.814 ± 0.005
Sniffing	Music	0.822 ± 0.008	0.790 ± 0.008	0.805 ± 0.008
Sniffing	Conversation	0.818 ± 0.008	0.785 ± 0.008	0.801 ± 0.008
Speech	Silence	0.652 ± 0.008	0.599 ± 0.007	0.624 ± 0.008
Walking	Silence	0.548 ± 0.005	0.494 ± 0.005	0.519 ± 0.005
Running	Silence	0.456 ± 0.004	0.309 ± 0.003	0.368 ± 0.003

Table 5.3: Accuracy of S_1 heart sound localisation by the segmentation algorithm.

The quality of the labelling generated by the segmentation algorithm is assessed by evaluating the precision and recall of the predicted S_1 states, relative to the ground truth ECG. If the *start* of the S_1 state occurs within 100ms of an R Peak in the corresponding ECG signal, then the segmentation algorithm is deemed to have correctly predicted the S_1 state. This evaluation metric was chosen so that it is possible to compare to Springer et al. [54]. Figure 5.4 provides an illustration of the segmentation against the ECG ground truth; during rest, the segmentation corresponds almost perfectly to the ground truth.

The results given in table 5.3 during rest are comparable to the results from [54] (F1 0.956)—despite the recordings in that study coming from digital stethoscopes, and being based on a more computationally expensive set of features. For deep breathing, there is a large difference between the precision and recall than was observed for other activities. This supports the hypothesis that the algorithm has difficulty making out heart sounds against breathing sounds.

There was significant inter-person variance during walking: two participants had F1 scores for segmentation of approximately 0.9. In the audio for these two participants it was found that the noise associated with the motion was below 50–60Hz, rather than the typical 100Hz. It appears that results during walking are affected by the weight of the footsteps, the exact positioning of the device, and the quantity of body fat in the region where the wearable is mounted: more fat means that vibrations have greater power, and take longer to dissipate, making inference more difficult. Over-sensitivity to device positioning and body characteristics is an undesirable flaw of the hardware, and must be considered when building the next iteration of hardware.

Despite the poor results for segmentation during walking, the heart rate estimates are reasonable. This behaviour is the result of the emission probability estimates for footsteps being high enough that the Viterbi algorithm confuses them for a heart sound, as there is a plausible time difference between the footstep and an actual heart sound. This leads to the true S_1 sound being reported as the S_2 sound, and the S_2 sound not being identified due to it having lower power than the S_1 sound. When the motion noise frequencies are suppressed to lower frequencies this failure mode does not occur, meaning high quality segmentation

is retained.

Heart Rate Variability

Activity	Noise Regime	Median Absolute Error / ms	Mean Absolute Error / ms	Median Percentage Error	Mean Percentage Error
Rest	Silence	4.01 ± 0.79	8.28 ± 1.30	6.14 ± 1.19	14.51 ± 2.66
Rest	Music	7.50 ± 6.10	19.18 ± 2.77	14.74 ± 9.37	39.25 ± 10.13
Rest	Conversation	7.78 ± 3.27	9.81 ± 1.43	13.26 ± 4.75	20.38 ± 3.10

Table 5.4: Accuracy of segmentation-based heart rate variability estimation algorithm.

Finally, the heart rate variability (HRV) accuracy was assessed. As HRV requires accurate segmentation, the results are only reported when the participant was resting. To the best of the author's knowledge, no empirical survey has assessed HRV accuracy for commercially available heart rate monitors, hence it is difficult to compare to commercial solutions. However, the errors are typically smaller than the 10ms resolution provided by the labelling. Values reported for HRV in healthy adults had an inter-quartile range of approximately 30ms [25]; the accuracy obtained by this approach is therefore likely sufficient for indicative readings.

5.3.2 Power Consumption and Latency

It has been shown that the audio data collected using a wearable device can be used to obtain accurate heart rate estimates, alongside heart rate variability estimates. The following section will demonstrate that the algorithms used to derive these estimates can be run *on-device*.

Benchmark Hardware

An STMicroelectronics Nucleo L496ZG-P development board [43] was used to run latency and power consumption experiments. This board featured a ARM Cortex-M4F processor running at 80MHz, with 320KB of SRAM and 1MB of flash. The board incorporated an external switched-mode power supply (SMPS); without this, the microcontroller's voltage regulator dominates the power consumption. This choice is representative of hardware that a real design could incorporate.

To measure latencies, the clock cycle register is used. Power measurements are reported for the microcontroller alone; measurements made for the entire board are dominated by status LEDs. It is worth noting that a contact microphone is a passive sensor, and does not need external power to make readings. Power consumption associated with the sensor will be from amplification and sampling, which is small relative to the microcontroller. The microcontroller current draw was measured at 4.5mA during run-mode and 276 µA during sleep; the voltage was 3.3V. Measurements assumed that ADC sampling at 11kHz, 16-bit was enabled continuously; reducing sampling rate to 500Hz reduced the current draw measured by approximately 1 µA. As these power measurements will be reused in the next chapter, where a higher sampling frequency is required, the results for the higher sampling rate are assumed.

Estimation

Algorithm 1 is sufficiently simple that a real-world implementation could be heavily optimised. One optimisation would be to push the filtering into analogue hardware. Designs could also push computation onto an efficient DSP processor, rather than relying on a microcontroller.

On the board selected, it takes 1802 µs to calculate a 1024 point FFT at the maximum precision supported; 1024 points is sufficient to fit 2 seconds of audio satisfying the Nyquist limit, as the maximum frequency of interest is 175Hz. As the algorithm calls for the computation of the Hilbert transform and the

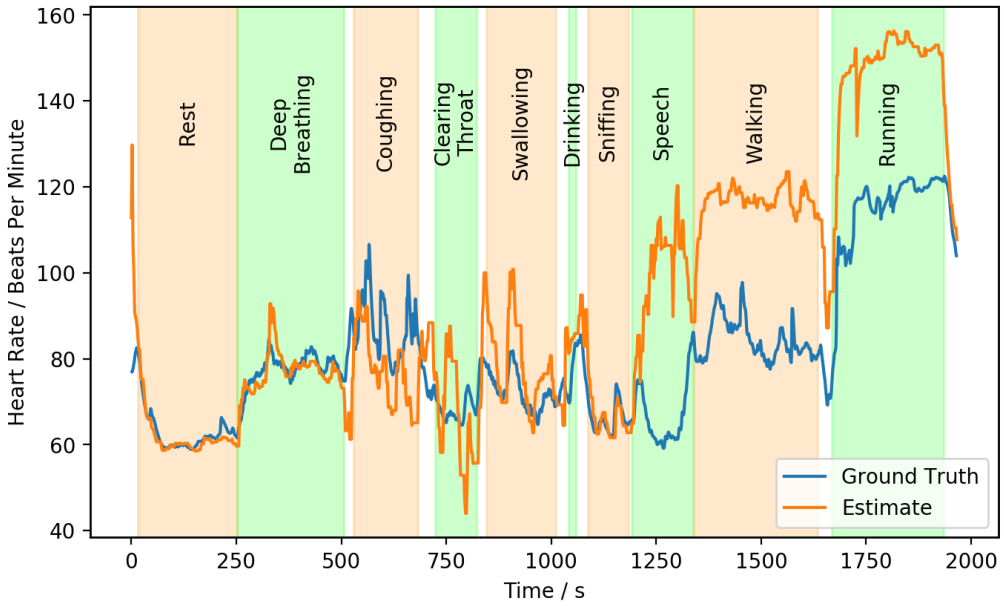
autocorrelation, it is necessary to perform 4 $\times$ 1024 point FFTs. Conservatively assuming that the rest of algorithm and other overheads causes an overall runtime of 10ms every 2 seconds, then the power consumption is under 1mW. Assuming a 3.7V, 100mAh lithium-polymer battery and a safety margin of 0.7, the battery life is 259 hours—*over a week*.

Segmentation

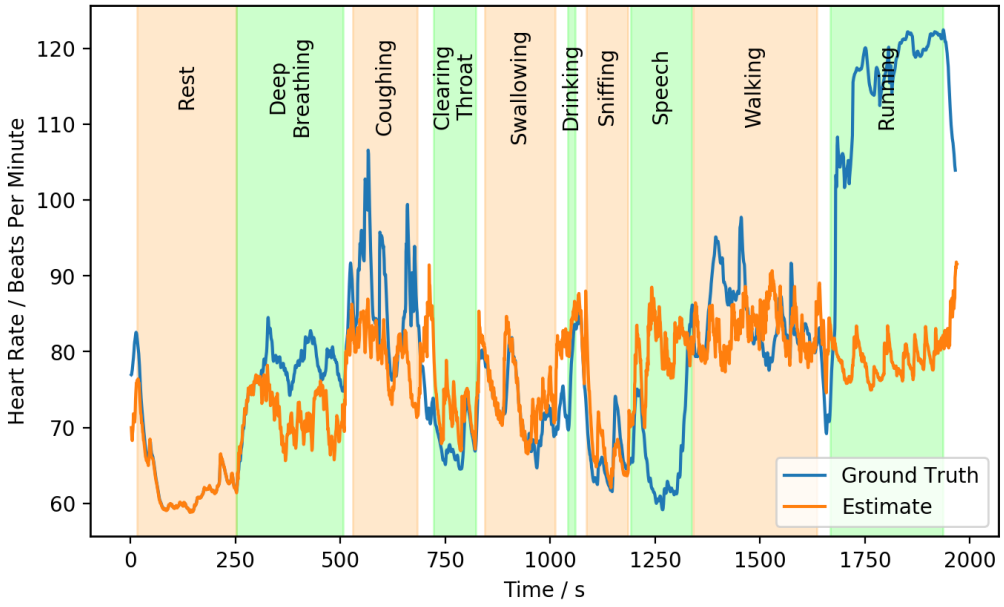
For segmentation, there are three contributors to the runtime:

1. Feature extraction. Only a 16 point RFFT is needed, but ARM's optimised libraries [10] do not support sizes below 32 points, which takes 29.1 µs. This means it takes approximately 3ms to extract 1 second of audio features.
2. Emission probability calculation. Inference for one set of features with random forest took 930 $\pm$ 56 cycles. This means it takes approximately 1.2ms to calculate 1 second of emission probabilities.
3. Viterbi algorithm. For 30s of data it took 20.86s to run the algorithm.

Assuming overheads, the segmentation algorithm requires approximately 700ms of computation for 1s of audio. Under the same battery assumptions, this yields a battery life of 24 hours.



(a) Autocorrelation



(b) Segmentation Method

Figure 5.3: Heart rate tracking accuracy for the two algorithms proposed for a single participant.

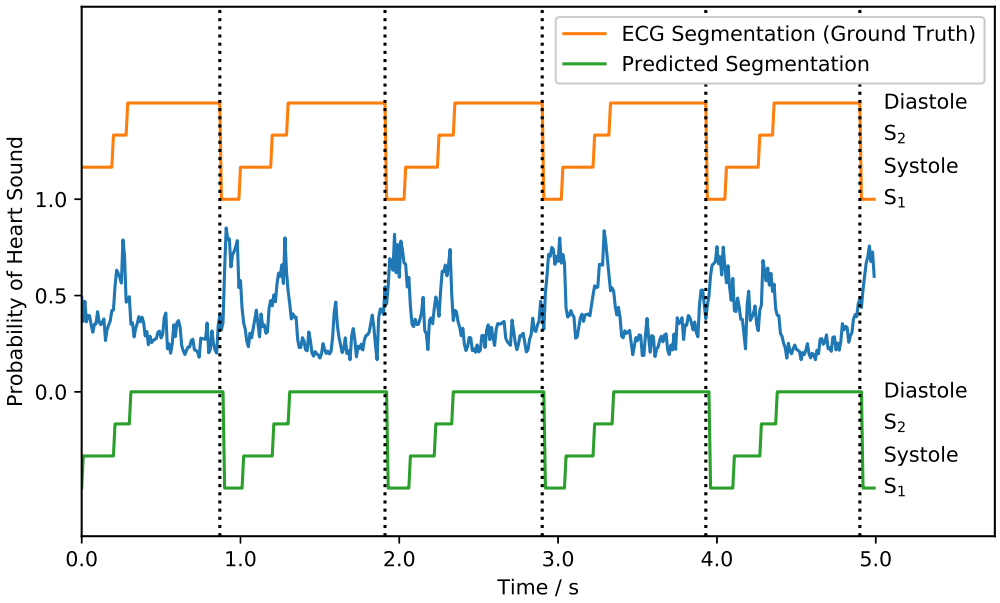


Figure 5.4: Predicted segmentation by the proposed algorithm compared to the ECG ground truth, when user is at rest. The emission probability used by the HSMM are also marked.

Chapter 6

Real-Time Asthma Monitoring

In this section the viability of *continuous, on-device* asthma monitoring will be explored. As asthma kills hundreds of thousands of people each year [15], construction of a device which can monitor symptoms and warn the wearer before they are aware of an issue could be of real value to society. A different data set was used to perform these experiments, due to the difficulty of obtaining my own dataset. Collecting data from individuals with specific medical conditions can take several years.

6.1 ICBHI Challenge 2017 Dataset

The largest publicly available data set for respiratory sounds was used to develop algorithms for continuous wheeze detection. The data set [50] was collected by two teams of researchers over several years under clinical conditions using electronic stethoscopes. It consists of 920 recordings from 126 patients, with recordings varying in placement. There are a total of 1392 wheeze events that have been annotated by a domain expert.

Although the data was collected using stethoscopes, it is argued that the techniques developed transfer to audio collected using a wearable device. As shown in section 4.3, it is possible to obtain high-quality recordings on a wearable device—

even with first revision hardware. It is also worth noting that the dataset was collected from several positions on the chest: the locations did not always correspond to that used by this project's data collection. In practice this is a useful test; robustness to positioning is a desirable feature of a classification algorithm.

Care is taken to split the dataset so that each individual is only represented in either the training or the test set. This prevents classification scores from being inflated by the classifier learning to identify traits related to individuals, and which are not related to the classification problem.

6.2 Model Selection

As mentioned in section 2.2.4, it is currently difficult to run recurrent networks on-device. In contrast, computer vision techniques are well supported. As a result, it is necessary to convert the audio so that the problem becomes one of computer vision. A popular technique for audio classification is to convert the audio to a spectrogram, and then treating it as an image [19]. This approach is the one adopted in this chapter.

6.2.1 Appropriate Input Representation

In order to find the optimal input representation, a search over various spectrogram parameters was done. Windows of audio were converted to (power) Mel spectrograms of size 32×32; the window length and maximum frequency was varied. There is a trade-off: a larger window size provides more context, but reduces temporal resolution. A similar argument can be made for the maximum frequency: a greater maximum frequency allows for the detection of more harmonics, but with lesser resolution.

To avoid overfitting, the stride used to extract windows of audio from the source was half the window size. As there are few wheezing examples, the stride is reduced to a quarter of the window size during wheezes, for the training data only. The size chosen was a compromise between resolution and runtime: convolutional

layers dominate the inference runtime, hence it is necessary to minimise the input size. A baseline network, using a residual architecture [18], was trained on each type of spectrogram for 40 epochs, using the Adam optimiser [27], with a learning rate of 10^{-4}; a cross entropy loss was used. To reduce overfitting, dropout is used. Each experiment was run 5 times against a validation set.

The results are given in fig. 6.1. Cohen's kappa [11] (κ) is used as the metric: it is more robust than accuracy as it considers the possibility of agreement by chance. Another benefit is that it allows us to compare the classifier to human classification, as it is used for evaluating inter-rater agreement. κ of 0.4, 0.6 and 0.8 correspond to moderate, substantial and almost-perfect agreement, respectively [30]. Wheeze detection is a difficult task: in one study [6], senior respiratory physicians obtained a (Fleiss) κ[1] of 0.54, while junior doctors obtained a *negative* κ. From this perspective, the best performance observed, 0.470 ± 0.047 compares favourably. This value was obtained at 1000Hz maximum frequency, which is sufficient to capture the highest fundamental frequencies observed for wheezes [40]. The window size of 1 second provides greater context, which helps the classifier to ignore noise at the same frequencies of interest. Too large a window, however, results in a wheeze being indistinguishable in the spectrogram.

6.2.2 Choosing a Model for On-Device Inference

A MobileNetV1 [22]-inspired architecture was chosen for running on-device. MobileNets were proposed as an efficient architecture for machine vision tasks (e.g. ImageNet classification), and are efficient in terms of the number of parameters and in latency: both aspects are vital when considering microcontroller applications, as they have limited storage space and computational resources. MobileNetV1's efficiency is achieved by replacing normal convolutional layers with *depthwise separable* convolutions. Normal convolutions are expensive as they operate on all channels in the input feature map; i.e. the cost of a convolution filter at a single spatial location involves $D_K \times D_K \times C$ multiply-add

[1] This is a multi-rater generalisation, but the same interpretation applies

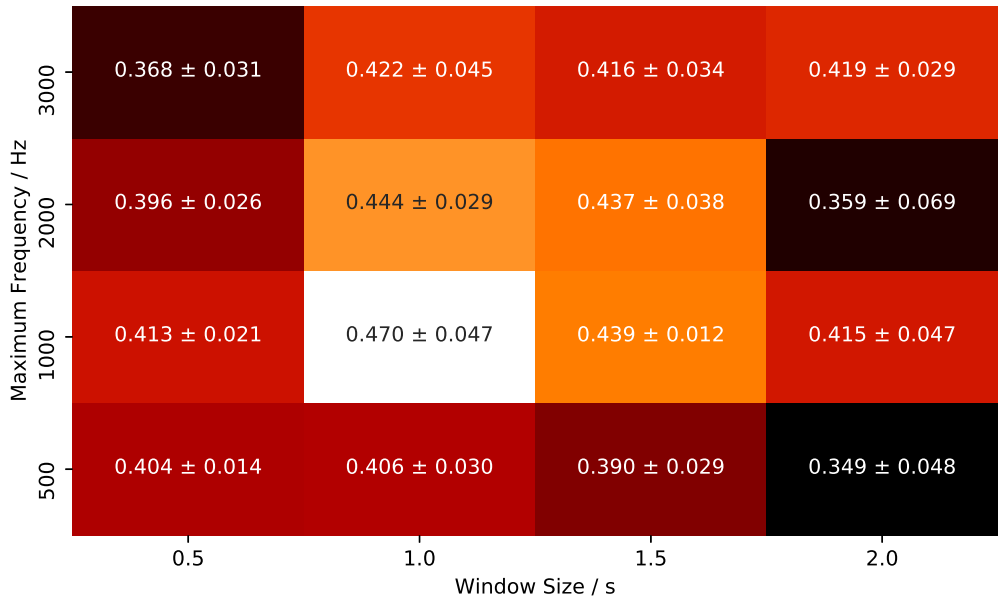
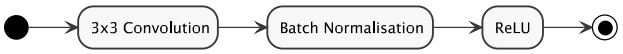
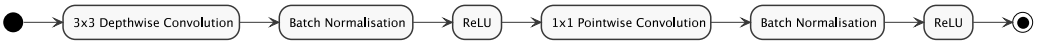


Figure 6.1: Results for the input representation search; values reported are Cohen's κ.

operations, where D_K is the kernel dimension and C is the number of channels in the input feature map. Depthwise separable convolutions consist of two distinct operations. The first phase applies depthwise filters that only interact with a single channel in the input feature map. A pointwise convolution—a convolutional layer with a 1×1 kernel—is used to create linear combinations of the feature maps generated by the depthwise convolutional layer. The exact setup, with the location of batch normalisation and non-linearities, is shown in fig. 6.2. It can be shown that depthwise separable convolutional layer uses $\frac{1}{D_K^2}$ multiply-add



(a) Standard convolution block.



(b) MobileNetV1 convolution block

Figure 6.2: Comparison of convolution layers in a normal convolutional neural network and a MobileNet [22].

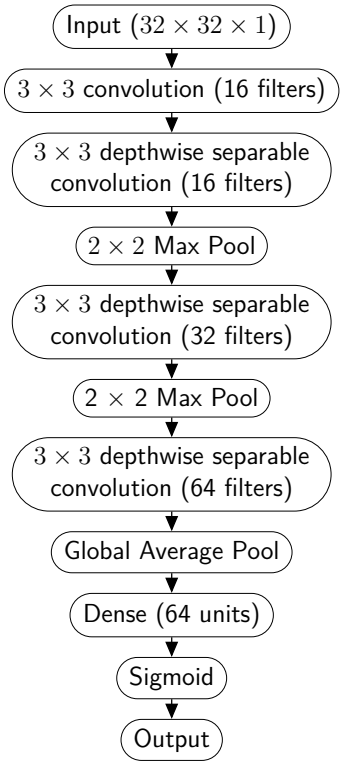


Figure 6.3: Architecture used for asthma detection on-device

operations of the equivalent normal convolutional layer.

The final model architecture used is shown in fig. 6.3. The network uses several depthwise separable convolutional blocks, with the exception of the first layer; as the input is a single channel, there is no difference. This architecture contains fewer than 7000 parameters, and other than being shallower than the original architectures proposed for ImageNet, the only other change made was to use max pooling. It was found that the original technique for downsampling by increasing the stride yielded worse results, likely because the convolutional blocks are not repeated several times before downsampling.

6.3 Evaluation

To evaluate the architecture's performance on device, the network was manually ported to the microcontroller development board described in section 5.3.2 using ARM's CMSIS-NN library [29]. This library contains optimised implementations of the operators used by the proposed architecture which exploit the SIMD instruction set of the CPU being used for the evaluation.

Optimised implementations do not work with floating point numbers. Instead, weights are quantised to fixed-point representations: each weight is represented with 8 bits. This procedure can cause accuracy degredation, but using 8 bits reduces the storage space required for the model and allows for the SIMD instructions to be executed 4-wide, reducing latency. Results are reported using the quantisation scheme used by CMSIS-NN [29]. The reader should note that the model used for inference does not contain batch normalisation layers: they can be folded into the preceding convolutional layers [24]. Quantisation is applied after the folding.

Network	**Test κ**	**8-bit Quantised Test κ**
1	0.460	0.448
2	0.485	0.479
3	0.499	0.505
4	0.483	0.469
5	0.485	0.491

Table 6.1: Results obtained on the test set for the on-device architecture

The architecture selected was trained on the optimal representation determined earlier in the chapter, and evaluated on the test set. The κ values obtained are reported in table 6.1; results before and after quantisation are given. The final κ values achieved by the quantised networks (mean 0.478) are in a human-level range [6].

The architecture's performance on-device was also evaluated. The latency of the network on the microcontroller was found to be 131.4ms. To compute the spectrogram, it is necessary to perform 32×512 point RFFTs; using ARM's DSP

library, this procedure was found to take 17.0ms. There is a small overhead to convert the STFT coefficients into the Mel basis; once this overhead is included, inference time can be conservatively estimated at 160ms of computation for 1s of audio, which comfortably enables real-time inference. Using the power measurements and battery assumptions in section 5.3.2, a runtime of *over 3 days* is obtained.

Chapter 7

Conclusion

This work has assessed the viability of a wearable device that continuously monitors non-speech body sounds collected from the user, to great success. The dissertation has considered the difficult challenges associated with wearables: *noisy measurements*, due to devices being used in the wild, and *limited energy and computational resources*. It has been shown that audio can be used to continously monitor the heart, even when the user is in noisy environments. Not only is it possible to obtain accurate heart rate estimates, with median percentage errors as low as 0.35% when the user is resting, it is also possible to obtain heart rate variability estimates. It was shown that these algorithms could be run on plausible hardware, with battery life of over a week for the cheaper algorithm. Real-time asthma symptom detection with near-human levels of classification was shown to be possible, with a battery life of over 3 days.

This project lays strong foundations for a multi-year project in the Mobile Systems Group: it is clear from these results that continuous monitoring of body sounds with wearables is viable. The techniques presented in this dissertation are sufficiently general that they can be adapted to suit future iterations of the hardware, and a dataset has been collected that will allow for a variety of problems to be explored; examples include respiratory cycle detection and respiratory event detection. Most importantly, this work has provided new insight into challenges that must be solved. These contributions are vital for informing future research

in this area.

7.1 Future Work

There are several directions for future work:

1. Construction of a new device. There are several aspects that could be revisited:
   (a) Improved sensors: it may be possible to build lightweight sensors that can be attached to the skin, without requiring an elastic strap. The benefit of this type of sensor is that it enables placements that were not achievable with the hardware in this project.
   (b) Integration of multiple sensors into the device. This would allow for the monitoring of several locations on the device, which could provide phase information. Exploitation of phase would be a novel research direction.
2. Respiratory event monitoring on-device. This dissertation explored wheeze detection using convolutional neural networks that ran on-device; in principle, this technique can be transferred to detecting coughing or sniffing.
3. Multi-modal data fusion. The hardware used for data collection included an IMU. One application where data could be combined would be respiratory cycle detection. Uni-modal approaches have been explored [72, 23] but, to the best of the author's knowledge, no approach has combined both.

Bibliography

[1] *A Python Toolbox for Statistics and Neurophysiological Signal Processing (EEG, EDA, ECG, EMG...).: Neuropsychology/NeuroKit.Py*. École de Neuropsychologie. URL: https://github.com/neuropsychology/NeuroKit.py (visited on 05/24/2019).

[2] *Amazon Alexa*. URL: https://developer.amazon.com/alexa (visited on 11/24/2018).

[3] Hamilton Bailey et al. *Bailey & Love's Short Practice of Surgery*. CRC Press, 2008. 1530 pp.

[4] D. Banerjee et al. "A Deep Transfer Learning Approach for Improved Post-Traumatic Stress Disorder Diagnosis". In: *2017 IEEE International Conference on Data Mining (ICDM)*. 2017 IEEE International Conference on Data Mining (ICDM). Nov. 2017, pp. 11–20.

[5] P Bifulco and GD Gargiulo. "Monitoring of Respiration, Seismocardiogram and Heart Sounds by a PVDF Piezo Film Sensor". In: *20th IMEKO TC4 Symposium on Measurements of Electrical Quantities: Research on Electrical and Electronic Measurement for the Economic Upturn* (), p. 4.

[6] Dina Brooks and Jackie Thomas. "Interrater Reliability of Auscultation of Breath Sounds Among Physical Therapists". In: *Physical Therapy* 75.12 (Dec. 1, 1995), pp. 1082–1088.

[7] *Cancer Research's CharityIndex Scores Have Risen | YouGov*. URL: https://yougov.co.uk/topics/consumer/articles-reports/2017/06/16/cancer-researchs-charityindex-scores-have-risen (visited on 05/13/2019).

[8] Michael Chu et al. "Respiration Rate and Volume Measurements Using Wearable Strain Sensors". In: *npj Digital Medicine* 2.1 (Feb. 13, 2019), p. 8.

[9] *Cloud Powered 3D CAD/CAM Software for Product Design | Fusion 360*. URL: https://www.autodesk.com/products/fusion-360/overview (visited on 05/07/2019).

[10] *CMSIS Version 5 Development Repository*. Arm Software. URL: https://github.com/ARM-software/CMSIS_5 (visited on 05/26/2019).

[11] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1 (Apr. 1, 1960), pp. 37–46.

[12] *Core ML | Apple Developer Documentation*. URL: https://developer.apple.com/documentation/coreml (visited on 04/30/2019).

[13] M. Faurholt-Jepsen et al. "Voice Analysis as an Objective State Marker in Bipolar Disorder". In: *Translational Psychiatry* 6 (July 19, 2016), e856. pmid: 27434490.

[14] Garmin and Garmin Ltd or its subsidiaries. *HRM-Run*. URL: https://buy.garmin.com/en-GB/GB/p/530376 (visited on 05/07/2019).

[15] GBD 2015 Mortality and Causes of Death Collaborators. "Global, Regional, and National Life Expectancy, All-Cause Mortality, and Cause-Specific Mortality for 249 Causes of Death, 1980-2015: A Systematic Analysis for the Global Burden of Disease Study 2015". In: *Lancet (London, England)* 388.10053 (Oct. 8, 2016), pp. 1459–1544. pmid: 27733281.

[16] Bernard J. Gersh. *Mayo Clinic Heart Book, Revised Edition: The Ultimate Guide to Heart Health*. Subsequent edition. New York: William Morrow, Jan. 15, 2000. 416 pp.

[17] Stephen Gillinov et al. "Variable Accuracy of Wearable Heart Rate Monitors during Aerobic Exercise". In: *Medicine & Science in Sports & Exercise* 49.8 (Aug. 1, 2017), pp. 1697–1703. pmid: 28709155.

[18] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[19] Shawn Hershey et al. "CNN Architectures for Large-Scale Audio Classification". In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017.

[20] *HIGH SENSITIVITY PIEZO FILM SENSOR: MEAS CONTACT MICROPHONE | TE Connectivity*. URL: https://www.te.com/usa-en/product-CAT-PFS0013.html#mdp-tabs-content (visited on 04/30/2019).

[21] *Home - OnScale*. URL: https://onscale.com/ (visited on 04/30/2019).

[22] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: (Apr. 16, 2017). arXiv: 1704.04861 [cs].

[23] M. Igras and B. Ziółko. "Wavelet Method for Breath Detection in Audio Signals". In: *2013 IEEE International Conference on Multimedia and Expo (ICME)*. 2013 IEEE International Conference on Multimedia and Expo (ICME). July 2013, pp. 1–6.

[24] Benoit Jacob et al. "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT: IEEE, June 2018, pp. 2704–2713.

[25] S. W. M. Keet et al. "Short-Term Heart Rate Variability in Healthy Adults". In: *Anaesthesia* 68.7 (2013), pp. 775–777.

[26] Kathi J Kemper, Craig Hamilton, and Mike Atkinson. "Heart Rate Variability: Impact of Differences in Outlier Identification and Management Strategies on Common Measures in Three Clinical Populations". In: *Pediatric Research* 62.3 (Sept. 2007), pp. 337–342.

[27] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[28] Parveen June Kumar and Michael L. Clark. *Acute Clinical Medicine*. Elsevier Health Sciences, Jan. 1, 2006. 758 pp.

[29] Liangzhen Lai, Naveen Suda, and Vikas Chandra. "CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs". In: (Jan. 19, 2018). arXiv: 1801.06601 [cs].

[30] J. Richard Landis and Gary G. Koch. "The Measurement of Observer Agreement for Categorical Data". In: *Biometrics* 33.1 (1977), pp. 159–174.

[31] N. D. Lane and P. Warden. "The Deep (Learning) Transformation of Mobile and Embedded Computing". In: *Computer* 51.5 (May 2018), pp. 12–16.

[32] Nicholas D. Lane, Petko Georgiev, and Lorena Qendro. "DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments Using Deep Learning". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp 2015*. Sept. 2015.

[33] Eric C. Larson et al. "Accurate and Privacy Preserving Cough Sensing Using a Low-Cost Microphone". In: *Proceedings of the 13th International Conference on Ubiquitous Computing - UbiComp '11*. The 13th International Conference. Beijing, China: ACM Press, 2011, p. 375.

[34] Shih-Hong Li et al. "Design of Wearable Breathing Sound Monitoring System for Real-Time Wheeze Detection". In: *Sensors* 17.1 (Jan. 17, 2017), p. 171.

[35] Rui Liu et al. "Vocal Resonance: Using Internal Body Voice for Wearable Authentication". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (Mar. 26, 2018), pp. 1–23.

[36] Hong Lu et al. "StressSense: Detecting Stress in Unconstrained Acoustic Environments Using Smartphones". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*. The 2012 ACM Conference. Pittsburgh, Pennsylvania: ACM Press, 2012, p. 351.

[37] Patrick J. Lynch. *Chest Landmarks, for Radiography and Other Chest Imaging Techniques*. Dec. 23, 2006. URL: https://commons.wikimedia.org/wiki/File:Thoracic_landmarks_anterior_view.svg (visited on 05/11/2019).

[38] Madhero88. *English: Phonocardiograms from Normal and Abnormal Heart Sounds*. Mar. 6, 2010. URL: https://commons.wikimedia.org/wiki/File:Phonocardiograms_from_normal_and_abnormal_heart_sounds.png (visited on 05/24/2019).

[39] manitou48. *Some Proof-of-Concept Sketches and Results for Arduino DUE: Manitou48/DUEZoo*. URL: https://github.com/manitou48/DUEZoo (visited on 05/12/2019).

[40] N. Meslier, G. Charbonneau, and J-L. Racineux. "Wheezes". In: *European Respiratory Journal* 8.11 (Nov. 1, 1995), pp. 1942–1948.

[41] *MPU-9250 | TDK*. URL: https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/ (visited on 04/30/2019).

[42] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. "Contactless Sleep Apnea Detection on Smartphones". In: *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (Florence, Italy). MobiSys '15. New York, NY, USA: ACM, 2015, pp. 45–57.

[43] *NUCLEO-L496ZG-P - STM32 Nucleo-144 Development Board with STM32L496ZGTP MCU, SMPS, Supports Arduino, ST Zio and Morpho Connectivity - STMicroelectronics*. URL: https://www.st.com/en/evaluation-tools/nucleo-l496zg-p.html (visited on 05/20/2019).

[44] Jeffrey E. Olgin et al. "Wearable Cardioverter–Defibrillator after Myocardial Infarction". In: *New England Journal of Medicine* 379.13 (Sept. 27, 2018), pp. 1205–1215. pmid: 30280654.

[45] *OPA1692 SoundPlus™ Low-Power, Low-Noise, High-Performance Dual Bipolar-Input Audio Op Amp | TI.Com*. URL: http://www.ti.com/product/OPA1692 (visited on 04/30/2019).

[46] *Polar H10 | Heart Rate Monitor Chest Strap*. URL: https://www.polar.com/uk-en/products/accessories//h10_heart_rate_sensor (visited on 05/07/2019).

[47] *Power Consumption Benchmarks | Raspberry Pi Dramble*. URL: https://www.pidramble.com/wiki/benchmarks/power-consumption (visited on 04/30/2019).

[48] Kiran K. Rachuri et al. "EmotionSense: A Mobile Phones Based Adaptive Platform for Experimental Social Psychology Research". In: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing - Ubicomp '10*. The 12th ACM International Conference. Copenhagen, Denmark: ACM Press, 2010, p. 281.

[49] Tauhidur Rahman et al. "BodyBeat: A Mobile System for Sensing Non-Speech Body Sounds". In: *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '14*.

The 12th Annual International Conference. Bretton Woods, New Hampshire, USA: ACM Press, 2014, pp. 2–13.

[50] Bruno M. Rocha et al. "An Open Access Database for the Evaluation of Respiratory Sound Classification Algorithms". In: *Physiological Measurement* 40.3 (Mar. 22, 2019), p. 035001. pmid: 30708353.

[51] Jonathan Rubin et al. "Recognizing Abnormal Heart Sounds Using Deep Learning". In: (July 14, 2017). arXiv: 1707.04642 [cs].

[52] S. E. Schmidt et al. "Segmentation of Heart Sound Recordings by a Duration-Dependent Hidden Markov Model". In: *Physiological Measurement* 31.4 (Mar. 2010), pp. 513–529.

[53] *Siri*. URL: https://www.apple.com/siri/ (visited on 11/24/2018).

[54] D. B. Springer, L. Tarassenko, and G. D. Clifford. "Logistic Regression-HSMM-Based Heart Sound Segmentation". In: *IEEE Transactions on Biomedical Engineering* 63.4 (Apr. 2016), pp. 822–832.

[55] David B. Springer et al. "Robust Heart Rate Estimation from Noisy Phonocardiograms". In: *Computing in Cardiology, CinC 2014, Cambridge, Massachusetts, USA, September 7-10, 2014*. 2014, pp. 613–616.

[56] Vivienne Sze et al. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". In: *Proceedings of the IEEE* (2017).

[57] Morton E. Tavel. "Cardiac Auscultation: A Glorious Past–and It Does Have a Future!" In: *Circulation* 113.9 (Mar. 7, 2006), pp. 1255–1259. pmid: 16520426.

[58] *Teensy Audio Library, High Quality Sound Processing in Arduino Sketches on Teensy 3.1*. URL: https://www.pjrc.com/teensy/td_libs_Audio.html (visited on 04/30/2019).

[59] *TensorFlow Lite*. URL: https://www.tensorflow.org/lite (visited on 04/30/2019).

[60] Julian F. Thayer et al. "A Meta-Analysis of Heart Rate Variability and Neuroimaging Studies: Implications for Heart Rate Variability as a Marker of Stress and Health". In: *Neuroscience & Biobehavioral Reviews* 36.2 (Feb. 2012), pp. 747–756.

[61] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[62] Catrine Tudor-Locke et al. "Walking Cadence (Steps/Min) and Intensity in 21–40 Year Olds: CADENCE-Adults". In: *International Journal of Behavioral Nutrition and Physical Activity* 16.1 (Jan. 17, 2019), p. 8.

[63] *Tympan*. URL: https://tympan.org/ (visited on 04/30/2019).

[64] Lei Wang et al. "Unlock with Your Heart: Heartbeat-Based Authentication on Commercial Mobile Phones". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.3 (Sept. 18, 2018), pp. 1–22.

[65] Ping Wang et al. "Phonocardiographic Signal Analysis Method Using a Modified Hidden Markov Model". In: *Annals of Biomedical Engineering* 35.3 (Mar. 1, 2007), pp. 367–374.

[66] Wapcaplet. *Diagram of the Human Heart, Created by Wapcaplet in Sodipodi. Cropped by Yaddah to Remove White Space (This Cropping Is Not the Same as Wapcaplet's Original Crop)*. 2006-06-02, 07:02. URL: https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_(cropped).svg (visited on 05/12/2019).

[67] *Welcome to BioSPPy — BioSPPy 0.6.1 Documentation*. URL: https://biosppy.readthedocs.io/en/stable/ (visited on 05/22/2019).

[68] S. C. White et al. "Comparison of Vertical Ground Reaction Forces during Overground and Treadmill Walking." In: *Medicine and science in sports and exercise* 30.10 (Oct. 1998), pp. 1537–1542. pmid: 9789855.

[69] DanielChangMD revised original work of DestinyQx; Redrawn as SVG by xavax. *English: A Wiggers Diagram, Showing the Cardiac Cycle Events Occuring in the Left Ventricle*. Mar. 20, 2012. URL: https://commons.wikimedia.org/wiki/File:Wiggers_Diagram.svg (visited on 05/12/2019).

[70] Koji Yatani and Khai N. Truong. "BodyScope: A Wearable Acoustic Sensor for Activity Recognition". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*. The 2012 ACM Conference. Pittsburgh, Pennsylvania: ACM Press, 2012, p. 341.

[71] Ali K. Yetisen et al. "Wearables in Medicine". In: *Advanced Materials* 30.33 (June 11, 2018), p. 1706910.

[72] Ja-Woong Yoon et al. "Improvement of Dynamic Respiration Monitoring Through Sensor Fusion of Accelerometer and Gyro-Sensor". In: *Journal of Electrical Engineering and Technology* 9.1 (Jan. 1, 2014), pp. 334–343.

[73] *Zephyr™ Performance Systems | Performance Monitoring Technology*. URL: https://www.zephyranywhere.com/ (visited on 05/08/2019).