

# Extraction of Vital Signs from Vibration Signals

Bachelor Thesis  
Faculty of Science, University of Bern

submitted by  
**Ramon Giulio Näf**  
from Bern, Switzerland

Supervision:

<sup>1</sup>PD Dr. Kaspar Riesen & <sup>2</sup>Prof. Dr. Tobias Nef

<sup>2</sup>Lena Bruhin & <sup>2</sup>Michael Single

<sup>1</sup>Institute of Computer Science (INF)

University of Bern, Switzerland

<sup>2</sup>Gerontechnology and Rehabilitation Group, ARTORG Center for  
Biomedical Engineering Research  
University of Bern, Switzerland

## Abstract

Sleep monitoring is essential for health tracking and well-being assessment, but the discomfort caused by body-attached devices during sleep highlights the need for non-intrusive monitoring technologies. The aim of this thesis is to reconstruct and enhance the model proposed by Li et al. for estimating heart rate (HR) and respiratory rate (RR) from vibration signals in a non-intrusive manner. The model employs Ensemble Empirical Mode Decomposition for signal decomposition, Principal Component Analysis on selected Intrinsic Mode Functions, and Short-Time Fourier Transform to extract time-resolved HR and RR functions. A grid search was conducted to optimise the hyper-parameters. The model was evaluated using data from a short-term controlled experiment. For HR estimation, the model achieved a Mean Absolute Error of  $5.78$  bpm, a Mean Percentage Error of  $1.18\%$ , and a Mean Squared Error of  $53.64$  bpm. Cross-correlation analysis showed a weak mean correlation of  $0.22$  with baseline HR data. The statistical analysis indicated that while the mean HR values estimated by the model closely aligned with baseline measurements, the variance was significantly higher in the baseline data. Bland-Altman analysis revealed a proportional bias, likely due to rounding effects, and Quantile-Quantile plots suggested a distribution close to normal for the estimated HR, contrasting with the baseline data. However, the model was unable to produce any RR estimates, highlighting areas for further improvement. In addition to the model reconstruction, a review of related work identified traditional signal processing techniques, such as the Hilbert Transform and Local Maxima Statistics, alongside machine learning approaches like Bi-LSTM, U-Net, and K-means models, as potential avenues for enhancement.



# Acknowledgements

I would like to express my deepest gratitude to my advisors, Lena Bruhin and Michael Single, for their unwavering support, insightful guidance, and invaluable feedback throughout this research. I am also sincerely thankful to PD Dr. Kaspar Riesen and Prof. Dr. Tobias Nef for granting me the opportunity to undertake this thesis and for their trust in my abilities. Finally, I would like to express my deepest appreciation to my friends and family. Their unwavering support, love, and patience have been the foundation that made this work possible.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Materials and Methods</b>	<b>4</b>
2.1	Data Collection . . . . .	4
2.1.1	Data Collection Systems . . . . .	5
2.1.2	Experimental Procedure . . . . .	5
2.2	Data Pre-processing . . . . .	6
2.2.1	Normalisation, Resampling and Filtering of Vibration Signals	7
2.2.2	Alignment and Segmentation of Vibration and Baseline Signals	8
2.3	Heart Rate and Respiration Rate Estimation . . . . .	10
2.3.1	Intrinsic Frequency Extraction of Cardiac and Respiratory Activities Using EEMD and PCA . . . . .	11
2.3.2	Extraction of Heart Rate and Respiration Rate Using STFT	12
2.4	Data Analysis . . . . .	14
<b>3</b>	<b>Related Works</b>	<b>17</b>
3.1	Traditional Signal Processing Approaches for Vital Sign Extraction	17
3.2	Artificial Intelligence Approaches for Vital Sign Extraction . . . . .	18
<b>4</b>	<b>Results</b>	<b>20</b>
4.1	Heart Rate and Respiration Rate Estimation . . . . .	21
4.1.1	Intrinsic Frequency Extraction of Cardiac and Respiratory Activities Using EEMD and PCA . . . . .	21
4.1.2	Extraction of Heart Rate and Respiration Rate Using STFT	23
<b>5</b>	<b>Discussion</b>	<b>27</b>
<b>A</b>	<b>Code</b>	<b>30</b>
<b>B</b>	<b>Graphs</b>	<b>33</b>
<b>C</b>	<b>Statistical Results</b>	<b>37</b>

<b>D Use of AI-based Tools</b>	<b>38</b>
<b>Bibliography</b>	<b>41</b>

# Chapter 1

## Introduction

Sleep monitoring plays a vital role in health tracking, significantly contributing to the assessment of overall well-being and the identification of potential health risks. Traditionally, this monitoring is accomplished by tracking respiration rate (RR) using breathing apparatuses [1], and measuring heart rate (HR) through body-contact wearables like chest straps and wrist sensors [2]. However, these methods often cause discomfort or are neglected at bedtime due to their intrusive nature, particularly among older adults or patients [3]. A variety of wearable healthcare devices have been developed for continuous electrocardiogram (ECG) monitoring, catering to both patients and health-conscious individuals [2]. Yet, the need for constant body attachment makes these devices cumbersome, particularly during sleep, highlighting the demand for less obtrusive, more user-friendly monitoring technologies [3]. There is, therefore, a need for non-intrusive solutions that continuously monitor cardiac events during night periods.

Innovatively, biomedical vibration signals, such as seismocardiograms (SCG) and ballistocardiograms (BCG), which measure the micro-vibrations produced by heart beats, are analysed for human health assessment and monitoring [4]. However, BCG monitoring is limited by the necessity for constant patient contact, often requiring specific equipment such as mattresses, pillows, or chairs to accurately detect vital signs [5]. SCG monitoring similarly often relies on sensors, such as accelerometers [6] or geophones [7], that need to maintain direct contact with the patient. Geophones, while advantageous for their insensitivity to lower-frequency movements, making them well-suited for heartbeat monitoring, are less responsive to respiratory vibrations due to their lower frequency [7]. This limitation necessitates the use of signal amplification and filtering for accurate detection of respiratory signals.

To address this issue, geophone-based seismographs, such as the Raspberry Shake 3D, which electronically extending the range of their geophones to detect

low-frequency vibration signals [8], may be utilised. These devices are also highly sensitive, allowing for the detection of faint seismic events from a greater distance [8]. Consequently, HR and RR vibration signals can be detected through a standard mattress without the need for direct body contact or specialised equipment. This non-invasive approach not only enhances patient comfort during sleep monitoring but also has the potential to increase adherence in studies requiring continuous vital sign monitoring.

Various approaches exist for the extraction of vital signs from vibration signals. Traditional signal processing techniques, such as the Hilbert transform [9], and auto-correlation functions [7], have been used due to their effectiveness in signal analysis. In addition to these traditional methods, machine learning approaches have also gained prominence. Techniques such as k-means clustering [6], bidirectional long short-term memory networks (Bi-LSTM) [10–12], and U-Net architectures [13] have proven to be valuable tools in the detection of heartbeats and respiratory activity from vibration signals. These approaches offer advanced capabilities for accurately identifying and monitoring vital signs in complex signal environments.

A common theme across many approaches for vital sign extraction is the focus on event detection, such as identifying individual heartbeats [5]. However, these techniques are limited by the quality of the vibration signal and therefore, susceptible to noise [6]. Additionally, machine learning algorithms, often regarded as black-box models, can present challenges when used for medical diagnosis, as their decision-making processes may lack transparency [14, 15]. This underscores the necessity for methods that are either explainable machine learning models, in which every component is transparent and justifiable, such as the selection of specific hyper-parameters and the number of iterations required for convergence, or those that rely exclusively on traditional machine learning models or signal processing techniques. Such methods would reduce the uncertainty inherent in the reasoning processes of machine learning algorithms.

In response to these challenges, the method proposed by Li et al. offers a promising alternative by utilising frequency analysis within a morphological model [3]. According to Li et al. [3], heartbeats and respiration generate vibrations within the frequency ranges of 0.75 to 2.4 Hz and 0.13 to 0.75 Hz, respectively. To analyse these signals, a Band-pass Filter (BPF) is applied, followed by Ensemble Empirical Mode Decomposition (EEMD) to extract the Intrinsic Mode Functions (IMFs) that correspond to these frequency ranges [3]. Following this, Principal Component Analysis (PCA) is applied to extract the First Principal Component (FPC) of the relevant IMF groups, which represents the intrinsic frequencies of cardiac and respiratory activities [3]. The Fast Fourier Transform (FFT) is then utilised to extract

the dominant frequency from the FPC, which is subsequently converted from Hertz (Hz) to beats per minute (bpm).

The aim of this thesis is to reconstruct the algorithm proposed by Li et al. [3] for estimating HR and RR from vibration signals collected by a seismograph and to extend upon it by utilising Short-Time Fourier Transform (STFT) to extract a function over time for HR and RR estimations. Following the reconstruction, the algorithm is tested on vibration data collected from volunteers during an experiment while being measured by seismographs. This vibration data is supplemented by ground-truth information collected using a commercially available ECG monitor to ensure accuracy and reliability. Lastly, the thesis includes a review of current signal processing techniques and machine learning methods for non-invasive monitoring solutions via vibration signals, examining their benefits and drawbacks.

# Chapter 2

## Materials and Methods

This chapter outlines the complete processing workflow for the collected vibration data, as illustrated in Figure 2.1, encompassing the key stages of measurement, pre-processing, and feature extraction. The measurement stage involves acquiring raw vibration data, as well as ground-truth data. During pre-processing, the raw data is normalised, aligned by resampling to synchronise timestamps, and filtered by applying a Butterworth BPF (BBPF) to isolate relevant frequency ranges. Following this, feature extraction is performed, where specific characteristics such as HR and RR are derived using techniques such as EEMD, PCA, and STFT [3]. All procedures are implemented using Python 3.11.9.

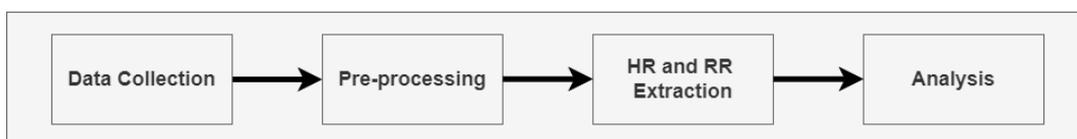


Figure 2.1: *Data Processing Pipeline*

### 2.1 Data Collection

A short-term controlled experiment was conducted with the primary aim of collecting vibration data of a participant’s HR and RR using a geophone-based seismograph. Simultaneously, the heartbeat of the participants was recorded with a commercially used ECG monitor to establish a ground truth. The experiment involved 10 participants, out of which 4 were female and 6 were male and were between 22 and 35 years old. The experiment was conducted at the Neurotec Loft at the ITEM Center for Biomedical Engineering Research in Bern.

### 2.1.1 Data Collection Systems

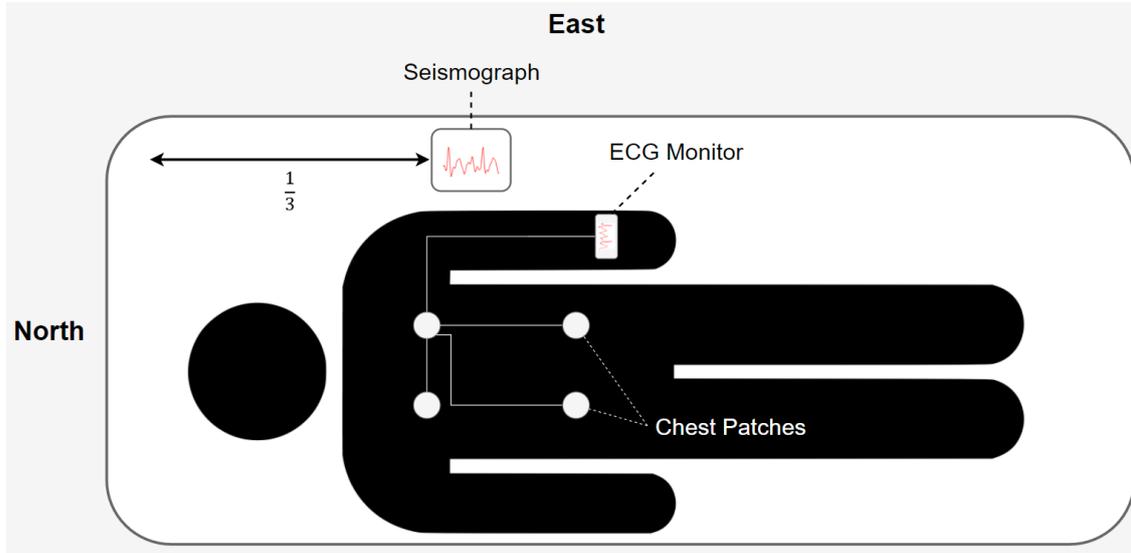


Figure 2.2: *Experimental setup showing a volunteer on a bed with a seismograph positioned to the east, one third south from the top of the bed. An ECG monitor is connected to the volunteer via chest patches. Note. The pictogram used in this image was sourced from svgsilh.com [16].*

As illustrated in Figure 2.2 a seismograph (RS-3D, Raspberry Shake S.A, Alto Boquete, Panama) was placed on a mattress and was used to record vibrations along three axes. The upwards-, north-, and east-axis, are referred to as EHZ, EHN, and EHE, respectively. During data processing, only the data from the EHZ-axis is ultimately considered as it contains information about the transverse motion. In the arrangement of the experimental setup, the head of the mattress was considered as the north direction. The seismograph was placed on the east side, one third south from top of the mattress. Furthermore, the EHZ-axis of the device was deliberately aligned to point upwards. This orientation ensured that the seismograph recorded the relevant vertical movements, thereby optimizing the accuracy and reliability of the data collected. For the recording of the vital sign ground-truth, a multimodal polygraph (SOMNOtouch<sup>TM</sup> NIBP, Randersacker, Germany) was used. This device was attached to the participants via four patches on the chest and lower stomach area, to record the participant's ECG.

### 2.1.2 Experimental Procedure

During the experiment, participants were instructed to lie on a bed in a supine position with their arms positioned at their sides. The data collection was organized into four stages to capture different variations in movement and respiration rate:

1. The participant maintains immobility while breathing at a respiratory rate between 8 to 45 bpm [3] for a duration of two minutes.
2. The participant is permitted to engage in slight movement for a duration of two minutes, encompassing actions such as rotating their head and executing random patterns of movement with their limbs.
3. The participant maintains supine immobility for two minutes and executed five consecutive deep inhalations, interspersed with ten-second intervals of normal respiration, until the prescribed duration elapses, or the last breath-set is completed.
4. The participant is instructed to synchronize their breathing artificially with a metronome set to a value between 8 to 45 bpm [3], according to the participant's comfortable range for a duration of two minutes.

A measurement of a participant took 15 to 20 minutes to complete. This included the introduction of the participants to the experiment process, the setup of the heartbeat monitor, and the collection of the vibration data. The heartbeat monitor was calibrated separately. During this process, the timestamps at both the initiation and conclusion of each stage were recorded. This ensured a comprehensive temporal record of the progression through the separate phases of the procedure. Furthermore, the quantity of inhalation sets in stage three and the respiratory rate in bpm in stage four were documented.

## 2.2 Data Pre-processing

The data pre-processing phase involves signal normalisation, alignment, and filtering to ensure consistency, comparability, and accuracy in the analysis. The raw data from both the seismograph and ECG monitor are first normalised to establish a uniform structure, allowing for coherent processing across different signal types. Then the vibration signal is filtered via the BBPF [3, 17], and resampled to avoid temporal gaps in the data. To assure accurate comparison, the vibration and baseline data are aligned based on their timestamps. Lastly, both signals are segmented to isolate the relevant time frames.



Figure 2.3: *Signal Filtering Using a 5<sup>th</sup> Order Butterworth Band-pass Filter*

## 2.2.1 Normalisation, Resampling and Filtering of Vibration Signals

The collected vibration data was normalised from raw data into a structured format, where each data point was assigned a timestamp. To ensure the continuity and integrity of the vibration data, the signal was resampled at a fixed interval of 10 milliseconds, wherein any missing values were forward-filled. This procedure was followed by the removal of any remaining null values. The implementation of this resampling process is detailed in Algorithm 1, where the resampling functionalities of Python library Pandas version 2.0.3 [18] were used. Following resampling, the vibration signal undergoes filtering using a 5<sup>th</sup> order BBPF (Figure 2.3). This choice is based on the methodology outlined by Shafiq et al. [17]. The filter is configured with a low cut-off frequency of 0.1 Hz and a high cut-off frequency of 8.0 Hz, as recommended by Li et al. [3]. This filtering process is implemented as shown in Algorithm 1, making use of the BBPF and filtering implementation of the Python library SciPy [19] version 1.13.1.

---

**Algorithm 1** Resampling and filtering of vibration data

---

```

from scipy.signal import butter, filtfilt

# formatted_df is the formatted data as a pandas DataFrame
resampled_df = formatted_df.resample('10ms').ffill()
resampled_df.dropna(inplace=True)
resampled_df = resampled_df.reset_index(drop=False)

ba = butter(N=5, Wn=[0.1,8.0], btype='band', output='ba', fs=100)
filtered_signal = filtfilt(ba[0], ba[1], resampled_df)

```

---

## 2.2.2 Alignment and Segmentation of Vibration and Baseline Signals

To ensure accurate comparison and analysis, the alignment of signals was achieved by resampling the baseline signal's timestamps to match those of the vibration signals. The process began by identifying the overlapping time spans of both signals, followed by cropping to retain only these common segments. Subsequently, the baseline signals were linearly interpolated to match the timestamps of the vibration signals, thereby aligning the datasets. This is demonstrated in Algorithm 2. Finally, the aligned vibration and baseline signals were segmented based on the time spans of the specific experiment stages, recorded during each measurement. These segments were then concatenated for subsequent analysis. The segmentation process was implemented as detailed in Algorithm 3.

---

**Algorithm 2** Alignment of vibration and baseline ecg signals

---

```
# Find overlapping period
start_overlap = max(vibration.index.min(), ecg.index.min())
end_overlap = min(vibration.index.max(), ecg.index.max())

# Truncate data to overlap
vibration = vibration[start_overlap:end_overlap]
ecg = ecg[start_overlap:end_overlap]
vibration.index = pd.to_numeric(vibration.index, errors='coerce')
ecg.index = pd.to_numeric(ecg.index, errors='coerce')

# Drop any NaN values that might have resulted
vibration = vibration.dropna()
ecg = ecg.dropna()

# Convert to numpy arrays and ensure they are 1-dimensional
x = np.asarray(vibration.index, dtype=np.float64).flatten()
xp = np.asarray(ecg.index, dtype=np.float64).flatten()
fp = np.asarray(ecg.values, dtype=np.float64).flatten()

interpolated_ecg = np.interp(x, xp, fp)

return pd.DataFrame(interpolated_ecg, index = vibration.index)
```

---

---

**Algorithm 3** Segmentation of a given vibration or baseline signal

---

```
# timeframes contains the timestamps of the start and end of  
# the experiment stages.  
stages = []  
idx = []  
  
# Find the timestamps closest to the start and  
# end time of the experiment stages  
for tf in timeframes:  
    differences = abs(np.subtract(timestamps, tf))  
    idx.append(np.argmin(differences))  
  
# Identify the indices of the relevant sections  
for i in range(0, len(idx), 2):  
    ts = timestamps[idx[i]:idx[i + 1]]  
    trunc_signal = signal[idx[i]:idx[i + 1]]  
  
# Calculate the duration in seconds  
dur = ts - ts[0]  
in_sec = dur.astype('timedelta64[ms]').astype(float) / 1000  
  
stage = pd.DataFrame({'timestamps' : in_sec,  
                     'signal': trunc_signal})  
stages.append(stage)
```

---

## 2.3 Heart Rate and Respiration Rate Estimation

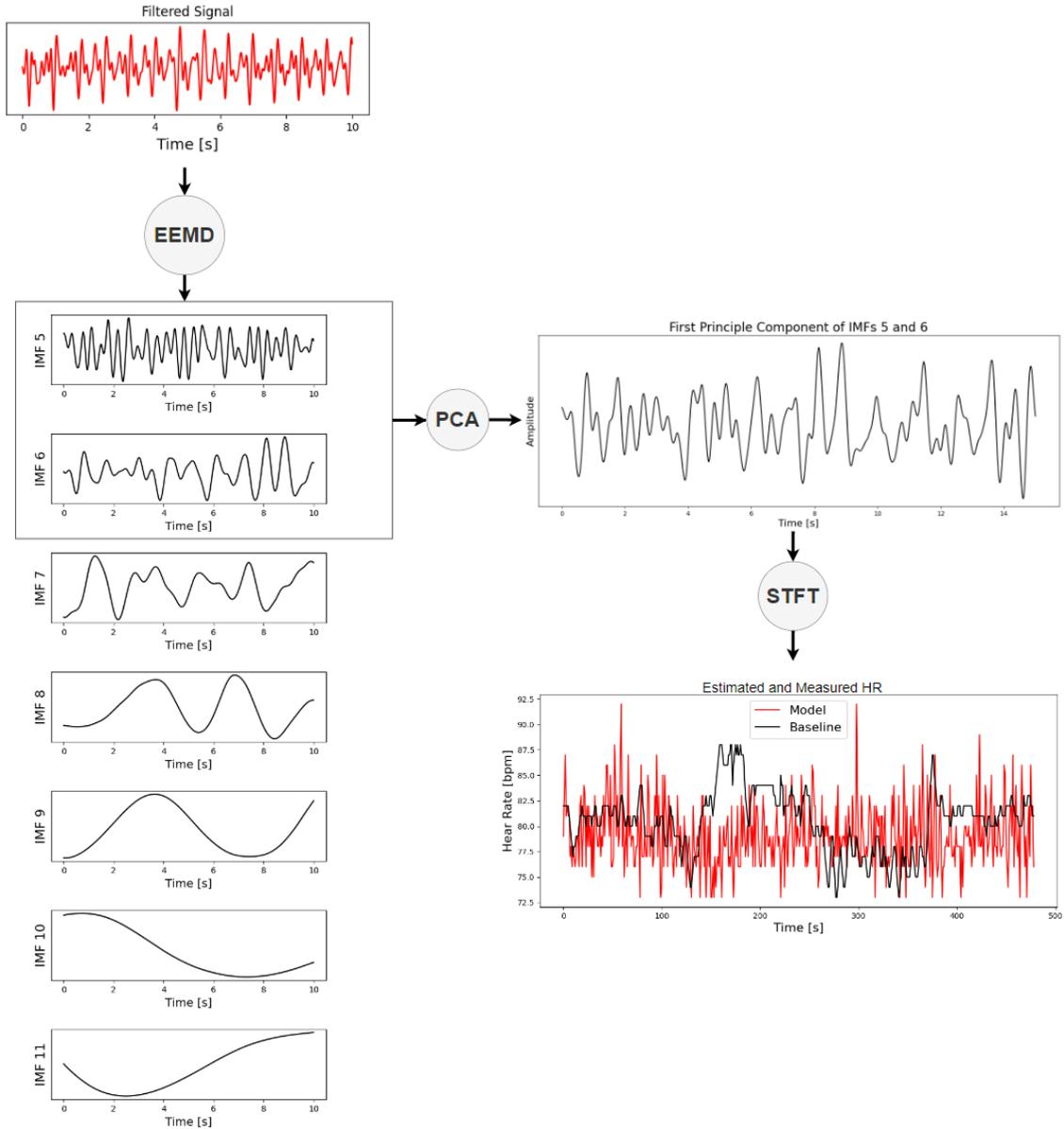


Figure 2.4: *The HR extraction process, including EEMD application to the filtered vibration signal, PCA on IMFs 5 and 6, and STFT on the resulting FPC.*

Figure 2.4 demonstrates the process of HR extraction. The RR extraction process is analogous. As recommended by Li et al. [3], the process began by identify IMFs 5 to 11 using EEMD. Then PCA was applied to the groups of IMFs 5 to 6 for HR and IMFs 7 to 11 for RR, extracting the First Principal Component (FPC) from each group [3]. The FPC was analysed using STFT to determine its dominant frequencies. By converting the dominant frequencies from Hz to bpm [3], a function over time for the HR and RR was obtained. Finally, a grid search, based on the Scikit-learn Python library [20], was used to determine the best hyper-parameters.

### 2.3.1 Intrinsic Frequency Extraction of Cardiac and Respiratory Activities Using EEMD and PCA

As suggested by Li et al. [3] the relevant frequencies were extracted from the vibration signals by identifying IMFs 5 to 11, using EEMD. The EEMD was computed using the implementation from the Python library PyEMD [21] version 0.2.13. To ensure repeatability, the number of trials per IMF candidate was set to 100, and the random noise seed was set to 131. Furthermore, to decrease computation time, the maximum number of calculated IMFs was limited to 11. The Python implementation can be seen in algorithm 4. Lastly, to validate, if the IMF groups covered the correct frequency ranges, spectrograms were generated for each IMF, using the STFT implementation from the Python library Scipy [19].

---

**Algorithm 4** Implementation of EEMD

---

```
from PyEMD import EEMD

eemd = EEMD(trials=100)
eemd.noise_seed(131)
eemd.eemd(S=signal, max_imf=11)
eIMFs, res = eemd.get_imfs_and_residue()
```

---

To extract intrinsic cardiac and respiratory information from the IMFs, PCA was applied to the HR and RR groups [3]. According to Li et al. [3], the FPCs present the intrinsic frequencies of cardiac and respiratory activities as they retain most of the variation in the selected IMFs. To extract the FPCs, as demonstrated in the presented code 5, the input array, comprised of the IMFs as column vectors, was first standardised using StandardScaler from the Python library Scikit-learn [20] version 1.5.0. The PCA implementation from the same library [20] was then applied to the standardised array, and the FPC was extracted.

---

**Algorithm 5** PCA implementation for extracting the FPC from IMFs 5 and 6

---

```
import numpy as np
import sklearn

# Format and standardise the IMF group
X = np.vstack((imf5, imf6)).T
scaler = sklearn.preprocessing.StandardScaler()
X_std = scaler.fit_transform(X)

pca = sklearn.decomposition.PCA(n_components=1)
principal_components = pca.fit_transform(X_std)
first_principal_component = principal_components[:, 0]
```

---

### 2.3.2 Extraction of Heart Rate and Respiration Rate Using STFT

To analyse the FPCs and extract a function over time for HR and RR, STFT was used. For this purpose, the STFT implementation from the Python library SciPy [19] version 1.5.0 was applied to the FPCs. To ensure that only the relevant frequencies were considered, the frequency array obtained from the STFT was truncated to specific ranges. For HR calculation, the frequency range was set to 0.75 to 2.4 Hz, while for RR, the range was restricted to 0.1 to 0.75 Hz. The dominant frequencies within these ranges were then extracted by calculating the maximum magnitudes of the truncated frequencies. The found dominant frequencies, expressed in Hz, were then converted to bpm by multiplying by 60. The Python implementation can be seen in Algorithm 6. To improve the models accuracy, the parameters for STFT were optimised using a custom grid search as seen in Algorithms 7, 8, and 9. The custom grid search is based on the Scikit-learn Python library [20] implementation.

---

**Algorithm 6** STFT implementation for the extraction of HR and RR

---

```
import numpy as np
from scipy.signal import stft

# Apply STFT
f, t, Zxx = stft(first_principal_component,
                 fs=100,
                 nperseg=win_len,
                 nfft=nfft,
                 noverlap= noverlap)

# Calculate the magnitude of all frequencies
magnitude = np.abs(Zxx)

# Truncate the frequencies to the frequency ranges for HR
min_freq, max_freq = 0.75, 2.4
frequency_mask = (f >= min_freq) & (f <= max_freq)
filtered_magnitude = magnitude[frequency_mask, :]

# Find the dominant frequencies
filt_peak_idx = np.argmax(filtered_magnitude, axis=0)
og_peak_idx = np.where(frequency_mask)[0][filt_peak_idx]
dominant_frequencies = f[og_peak_idx]

# Convert to BPM
HR = np.round(dominant_frequencies * 60)
```

---

---

**Algorithm 7** Grid search implementation

---

```
from sklearn.metrics import mean_squared_error

if is_HR:
    param_grid = {'win_len': list(range(10, 200, 2)),
                  'hop_size': list(range(0, 100, 2)),
                  'nfft': [256, 512, 1024, 2048, 4096, 8192],
                  'min_freq': 0.8, 'max_freq': 2.0}
else:
    param_grid = {'win_len': list(range(10, 75, 2)),
                  'hop_size': list(range(0, 45, 2)),
                  'nfft': [256, 512, 1024, 2048, 4096, 8192],
                  'min_freq': 0.1, 'max_freq': 0.75}

vse = VitalSignExtractor() # The created model
predictions, tested_params = [], []
averaged_predictions, mean_squared_errors = [], []

for win_len, noverlap, nfft, min_freq, max_freq in product(
    param_grid['win_len'],
    param_grid['noverlap'],
    param_grid['nfft']):

    if noverlap > win_len : continue
    if nfft <= win_len : continue
    params = { 'fs': 100, 'win_len': win_len,
               'noverlap': noverlap, 'nfft': nfft,
               'min_freq': param_grid['min_freq']
               'max_freq': param_grid['max_freq']
             }

    y_pred, avg_y_pred, mse = test_parameters(X, y, vse, params)

    if y_pred : # If the tested parameters were valid
        tested_params.append(params)
        predictions.append(y_pred)
        averaged_predictions.append(avg_y_pred)
        mean_squared_errors.append(mse)

# Find index of parameters with lowest MSE
idx = np.argmin(mean_squared_errors)
best_params = tested_params[idx]
best_score = mean_squared_errors[idx]
best_pred = predictions[idx]
best_averaged_prediction = averaged_predictions[idx]
vse.set_params(**best_params)
```

---

---

**Algorithm 8** Parameter test for the grid search implementation

---

```
vse.set_params(**params_to_test)
vse.fit(X, y)
y_pred = vse.predict(X)

y_pred = np.array(y_pred)
formatted_y_pred = format_for_comparison(y_pred, y)

# To avoid false constant HR estimation
if not all(x == formatted_y_pred[0] for x in formatted_y_pred):
    mse = mean_squared_error(y, formatted_y_pred)
    return y_pred, formatted_y_pred, mse

else :
    return None, None, None
```

---

---

**Algorithm 9** Averaging the Estimated Vital Sign Results

---

```
import numpy as np

ratio = len(predicted_pulse) / len(recorded_pulse_per_second)

return np.array([np.round(\
    np.mean(\
        predicted_pulse[\
            int(i * ratio):int((i + 1) * ratio)])\
        for i in range(len(recorded_pulse_per_second))])])
```

---

## 2.4 Data Analysis

This section provides a comprehensive overview of the model's analysis pipeline. The process begins with model validation, followed by a qualitative evaluation of the model's performance, and concludes with a descriptive statistical analysis of the results' characteristics. The evaluation approach is multifaceted, incorporating both quantitative and qualitative metrics to ensure a thorough assessment of the model's accuracy and reliability. Various statistical techniques are applied to measure the agreement between estimated and observed data, complemented by detailed visualisations to enhance the interpretability of the findings.

---

**Algorithm 10** Cross-validation implementation

---

```
import numpy as np

training_sets, test_sets = k_split(X, y, k)
mse_train_scores = []
mse_test_scores = []
best_models = []
best_params_list = []

for (X_train, y_train), (X_test, y_test) in zip(
    training_sets, test_sets):

    (model,
     pred_train,
     avg_pred_train,
     best_params_train,
     best_score_train) = custom_grid_search(X_train,
                                           y_train,
                                           param_grid)

    assert model.is_fitted_
    pred_test = model.predict(X_test)
    avg_pred_test = format_for_comparison(pred_test, y_test)

    mse_train = mean_squared_error(y_train, avg_pred_train)
    mse_test = mean_squared_error(y_test, avg_pred_test)

    mse_train_scores.append(mse_train)
    mse_test_scores.append(mse_test)
    best_models.append(model)
    best_params_list.append(best_params_train)

avg_mse_train = np.mean(mse_train_scores)
avg_mse_test = np.mean(mse_test_scores)

best_index = np.argmin(mse_test_scores)
best_params = best_params_list[best_index]
```

---

The model validation was conducted using techniques such as cross-validation, cross-correlation, mean absolute errors (MAE), mean percentage errors (MPE), and mean squared error (MSE). Cross-validation was employed to assess the model's ability to generalise to new data by dividing the dataset into multiple subsets and evaluating the model's performance on each subset. The python implementation can be seen in Algorithm 10. From the Python library Numpy [22] version 1.26.0,

the cross-correlation implementation was utilised to measure the similarity between the estimation and actual signals, providing insight into the temporal alignment and overall accuracy of the model. To assess the models accuracy, MAE and MPE were implemented using the before mentioned Numpy [22] Library, while the MSE implementation from the Scikit-learn [20] library was used.

The qualitative evaluation of the model's performance was carried out using several statistical metrics, including mean, standard deviation, variance, kurtosis, and skewness. The mean provides a measure of central tendency, while the standard deviation and variance quantify the dispersion of the data. Kurtosis and skewness were used to assess the shape and asymmetry of the distribution, offering deeper insights of the model's behaviour. To calculate these values, the implementations from the Python library Numpy [22] version 1.26.0 were used.

Descriptive statistical measures were employed to further analyse the model's performance and the characteristics of the data. Bland-Altman plots were used to visualise the agreement between the model's estimations and the actual measurements, while Quantile-Quantile (QQ) plots assess the normality of the data distribution. For both Bland-Altman- and QQ-Plots, the Python library Pingouin [23] version 0.5.4 was used. General signal visualisation techniques using the Python library Matplotlib [24] version 3.7.1 were also applied to provide an intuitive understanding of the data patterns and any discrepancies between the modelled and observed signals.

# Chapter 3

## Related Works

Recent advancements in monitoring technologies that utilise vibration signals have significantly expanded their application across various domains, including at-home, clinical, and vehicular environments. Traditional signal processing techniques have been extensively employed in this field, providing valuable insights into non-invasive monitoring methods. Additionally, several studies have introduced machine learning models for the extraction and analysis of cardiovascular signals, with a particular emphasis on non-invasive approaches. This chapter offers an overview of the methods used for extracting vital signs from vibration signals, encompassing both traditional signal processing techniques and machine learning approaches.

### 3.1 Traditional Signal Processing Approaches for Vital Sign Extraction

This section discusses various methodologies, highlighting their respective benefits and challenges. Jafari et al. [9] introduce a Hilbert adaptive beat identification technique for detecting heartbeat timings and inter-beat intervals from SCG using a tri-axial microelectromechanical accelerometer. This method demonstrates high correlation and accuracy across different body positions, making it suitable for real-time continuous cardiac monitoring. Furthermore, since it is based on Hilbert Transform, it is able to create a function over time for the HR. However, its performance is notably impacted by noise, particularly in low-quality signals or during dynamic movements, which presents a significant challenge in practical applications.

Similarly, Jia et al. [25] propose VitalMon, a geophone-based system designed to monitor heart and respiratory rates even when two individuals share a bed. Their system implements a combination techniques, such as amplitude modulation, square-law amplitude demodulation, auto-correlation functions, FFT, and blind

source Separation techniques. The system effectively separates and analyses mixed heartbeat signals, achieving low estimation errors for both heart and respiratory rates. Despite its accuracy and non-intrusiveness, VitalMon struggles with the detection of low-frequency respiratory signals, as geophones are not optimally sensitive to such vibrations. Moreover, monitoring heart rates in a shared bed environment remains challenging due to overlapping signals.

Shafiq et al. [17] present an automated approach for annotating SCG peaks required for calculating systolic time intervals, using a template matching approach combined with sliding templates and segmentation. This method is particularly advantageous for long-term cardiac health monitoring in wearable configurations, offering robust peak detection under various conditions. However, the method faces difficulties in seated trials, where motion artifacts complicate accurate signal annotation, necessitating further development to enhance its robustness against such disturbances.

Lastly, Clemente et al. [26] explore the use of a bed-mounted seismometer system to monitor heart and respiratory rates, body movement, and posture during sleep. Their approach combines local maxima statistics for HR detection and synchrosqueezed wavelet packet transform for RR estimation in a non-intrusive manner. Although the system is effective and cost-efficient, the commodity seismometer employed is insensitive to low-frequency measurements, limiting its ability to directly observe respiratory rates, which poses a challenge for comprehensive sleep monitoring.

## 3.2 Artificial Intelligence Approaches for Vital Sign Extraction

Several studies have proposed machine learning models for extracting and analysing cardiovascular signals, with a focus on non-invasive methods. For example, the system "HeartQuake" by Park et al. [10], explores the use of geophone-based systems to capture cardiac activity patterns without direct body contact. HeartQuake employs a Bi-LSTM model to estimate ECG signals from vibration data collected through a mattress. This method provides a low-cost, non-intrusive solution for accurate ECG waveform estimation, demonstrating its utility in both clinical and home settings. The system effectively captures key ECG components, proving its potential for large-scale deployment in remote patient monitoring [10]. Despite its low cost and non-invasiveness, the system faces challenges due to sensor noise and external vibrations, which necessitate advanced filtering techniques. The need for a

personalised model to mitigate errors under varying conditions also highlights the complexity of real-world deployment [10].

Furthermore, Chen et al. [11] proposed a beat-to-beat heart rate detection method also using a Bi-LSTM network, leveraging low-frequency SCG signals. Their approach involved training a regression model to predict ECG signals from SCG data, achieving high sensitivity and precision. This method facilitates continuous heart rate monitoring in a non-invasive manner, making it suitable for integration into wearable devices and daily living environments [11]. The direct use of raw SCG signals without feature extraction simplifies the process, but the variability of SCG waveforms across different subjects and activities poses significant challenges [11]. Additionally, the limited dataset used for model training suggests the need for further research on diverse signal directions [11].

Chan et al. [13] introduced a U-Net-based framework that estimates respiratory rates from ECG- and SCG-derived respiratory signals. This approach transforms the signals into the spectro-temporal domain, denoising them through a 2D U-Net to reduce non-respiratory artifacts and fuses multi-modal inputs, resulting in a low MAE of 0.82 bpm [13]. The generalisability of the model was validated with unseen respiratory rates, making it a robust tool for real-world applications. However, the requirement for resampling the demodulated respiratory signals for practicality poses a challenge, especially for real-time implementation [13].

Alongside these deep learning-based methods, López-Rico and Ramírez-Chavarría [6] employed a K-Means clustering algorithm for SCG data processing. Their approach automatically labels waveform events, offering a cost-effective solution for smart seismocardiography devices. Their study demonstrated the effectiveness of this method in accurately grouping cardiovascular events, which could enhance the development of home-made, smart health monitoring devices [6]. The clustering technique demonstrates excellent performance in grouping cardiovascular events with high accuracy [6]. However, the dependency on good-quality SCG signal morphology and the limitation to breath-holding scenarios indicate areas for further improvement and multivariable analysis to enhance the system's robustness [6].



# Chapter 4

## Results

### 4.1 Heart Rate and Respiratin Rate Estimation

#### 4.1.1 Intrinsic Frequency Extraction of Cardiac and Respiratory Activities Using EEMD and PCA

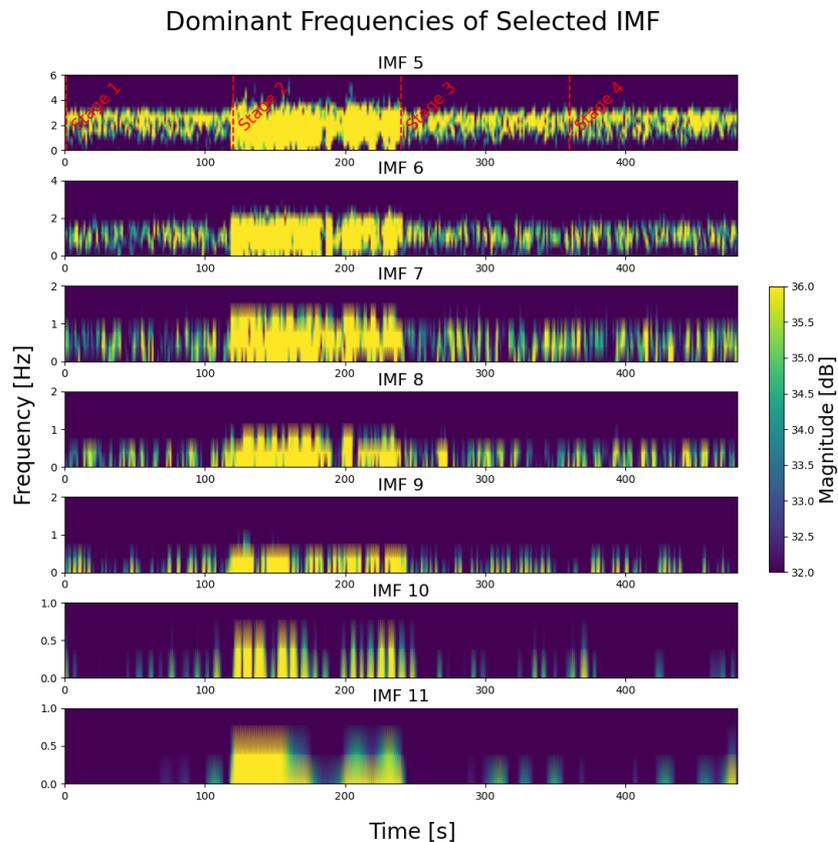


Figure 4.1: A spectrogram presenting the frequency spectrum of IMF 5 to 11. The red dotted lines mark the expereriment stages. Note. Only frequencies with a log magnitude multiplied by ten between 32 and 36 dB are visualised.

This section presents the frequency analysis of the IMFs found via EEMD. A spectrogram visualising the frequency ranges of of IMFs 5 to 11 is presented in Figure 4.1. The ranges of IMFs 5 to 7 for one participant achieved a maximum dominant frequency of 23.05 Hz, 28.52 Hz, and 14.45 Hz due to a peak at the end of the signal, and were considered outliers (Figure 4.2). The maximum dominant frequencies of IMFs 5 to 11, with and without outliers, can be seen in Table 4.1.

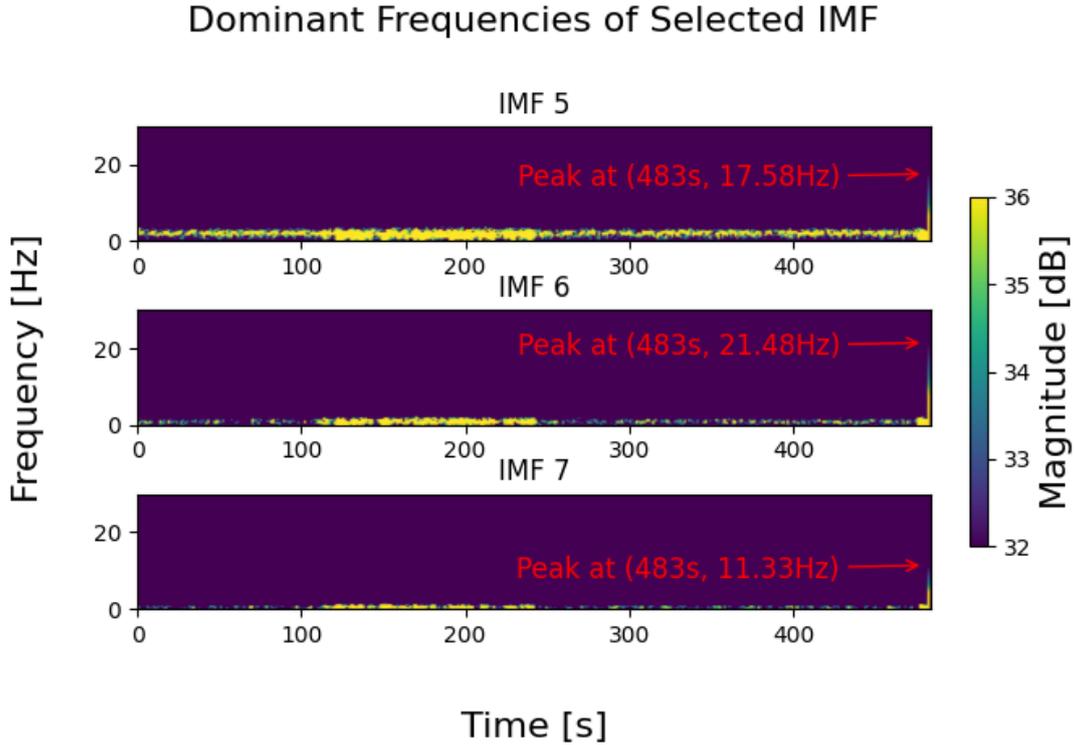


Figure 4.2: A spectrogram presenting the dominant frequency spectrum of IMF 5 to 7 considered to be outliers due to their peak at the end. Note. The peak of IMF 5, 6 and 7 are found at a frequency of 17.56Hz, 21.48Hz, and 11.33Hz, respectively.

Table 4.1: The average maximum dominant frequencies in Hz of IMFs 5 to 11, with and without outliers. Note. All IMFs reached a minimum dominant frequency of 0.0 Hz.

IMFs	Maximum (With Outlier)	Maximum (Without Outlier)
5	8.15	6.58
6	6.19	3.65
7	3.18	1.82
8	2.01	2.08
9	2.51	2.8
10	0.95	1.04
11	1.06	1.17

## 4.1.2 Extraction of Heart Rate and Respiration Rate Using STFT

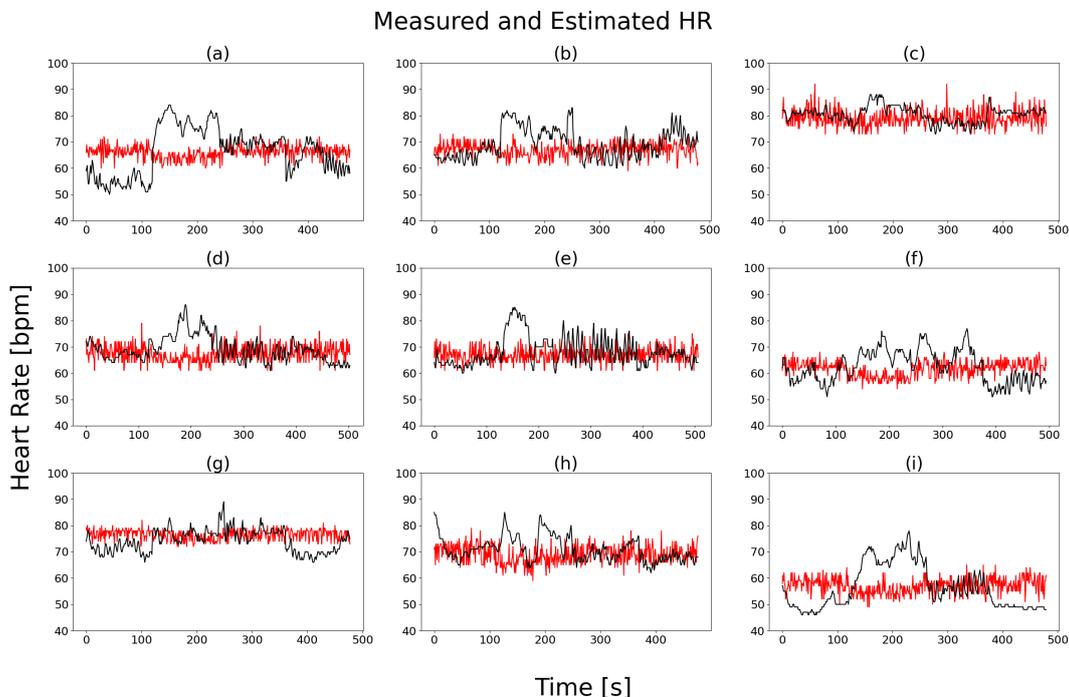


Figure 4.3: A comparison of the model’s estimated HR (red), compared to their baseline measurement HR (black) for each participant. Note. In each graph, the data from all experiment stages are shown.

The estimations of the model, compared to their baseline measurements can be seen in Figure 4.3. The model did not return any results for RR estimation. On average, for HR estimation the algorithm demonstrated a MAE of  $5.78$  bpm, with a MPE of  $1.18\%$ , and a MSE of  $53.64$ . Additionally, cross-correlation analysis revealed an average correlation of  $0.22$  in the range of  $-1$  to  $+1$ . Three-fold cross-validation of the model yielded an average MSE of  $27.53$  for the training sets and  $86.18$  for the test sets. In Table 4.2, the MSE score for the training and test sets can be seen for each individual participant. The hyper-parameter identified through the grid search for STFT are presented in Table 4.3. The parameter grid of possible parameters included the window length ranges from  $10$  to  $200$  with an overlap of  $0$  to  $100$  samples for HR, and  $10$  to  $75$  with an overlap of  $0$  to  $45$  samples for RR. The number of samples per FFT group was set to be between  $256$  to  $8192$  in steps of powers of  $2$ .

Table 4.3: *A listing of participant-specific hyperparameters used to optimise the STFT, identified via a grid search.* Note. The window length ranged between 10 to 18 samples, the overlap was between 8 and 12 samples, and the FFT group size varied between 512 and 4096 samples in powers of 2.

Participant	Window Length	Overlap	FFT Group Size
1	10	8	512
2	10	8	512
3	18	16	256
4	12	10	512
5	12	10	512
6	10	8	1024
7	10	8	256
8	14	12	512
9	10	8	4096

Table 4.2: *The highest MSE scores from three-fold cross-validation applied to the model for each participant of the experiment.* Note. Training MSE scores range from 18.4 to 53.73, while the test MSE scores are between 37.38 and 194.09.

Participant	MSE Train	MSE Test
1	23.35	194.09
2	20.51	115.7
3	18.4	40.36
4	20.97	65.01
5	24.12	74.14
6	36.48	49.03
7	20.52	37.38
8	29.72	60.64
9	53.73	139.27

To qualitatively evaluate the model’s performance, statistical metrics including the overall mean ( $M$ ), standard deviation ( $SD$ ), variance ( $s^2$ ), kurtosis ( $g_1$ ), and skewness ( $g_2$ ) for both the model and the corresponding baseline measurements are provided in Table 4.4. The same statistical metrics can be seen in Tables 4.5 and 4.6 for each individual participant of the experiment.

Table 4.4: *Overall statistical evaluation of the model’s estimated HR compared to it’s baseline measured HR.*

Statistic	Mean	Standard Deviation	Variance	Skewness	Kurtosis
Model	68.56 bpm	2.81 bpm	8.09 bpm	0.04	-0.15
Baseline	67.83 bpm	5.7 bpm	35.65 bpm	0.5	-0.24

Table 4.5: *A complete statistical evaluation of the estimated HR for each participant.* Note.  $M$  ranges from 57.01 to 79.05 bpm, with  $SD$  between 3.85 and 12.28 bpm, and  $s^2$  from 1.96 to 3.5 bpm. Furthermore,  $g_2$  values range from -0.32 to 0.69, and  $g_1$  values range from -0.83 to 0.79.

<b>Participant</b>	Mean	Standard Deviation	Variance	Skewness	Kurtosis
1	65.99 bpm	5.75 bpm	2.4 bpm	-0.15	0.06
2	67.04 bpm	6.64 bpm	2.58 bpm	-0.32	0.12
3	79.05 bpm	10.31 bpm	3.21 bpm	0.69	0.79
4	67.49 bpm	9.35 bpm	3.06 bpm	0.35	0.04
5	66.88 bpm	7.24 bpm	2.69 bpm	0.27	0.08
6	61.57 bpm	8.22 bpm	2.87 bpm	-0.2	-0.62
7	76.52 bpm	3.85 bpm	1.96 bpm	-0.32	-0.83
8	68.96 bpm	12.28 bpm	3.5 bpm	0.05	-0.54
9	57.01 bpm	9.16 bpm	3.03 bpm	0.03	-0.44

Table 4.6: *A complete statistical evaluation of the measured baseline HR for each participant.* Note.  $M$  ranges from 56.2 to 80.45 bpm, with  $SD$  between 9.44 and 81.0 bpm, and  $s^2$  from 3.07 to 9.0 bpm. Furthermore,  $g_2$  values range from -0.04 to 1.12, and  $g_1$  values range from -1.09 to 0.76.

<b>Participant</b>	Mean	Standard Deviation	Variance	Skewness	Kurtosis
1	66.19 bpm	81.0 bpm	9.0 bpm	0.0	-1.09
2	69.35 bpm	31.38 bpm	5.6 bpm	0.52	-0.68
3	80.45 bpm	9.44 bpm	3.07 bpm	-0.04	-0.04
4	69.13 bpm	23.27 bpm	4.82 bpm	0.85	0.58
5	68.26 bpm	30.96 bpm	5.56 bpm	1.12	0.76
6	62.37 bpm	36.73 bpm	6.06 bpm	0.23	-0.84
7	74.17 bpm	18.13 bpm	4.26 bpm	0.23	-0.39
8	70.88 bpm	21.16 bpm	4.6 bpm	0.81	0.27
9	56.2 bpm	68.79 bpm	8.29 bpm	0.76	-0.72

Two QQ plots evaluating the normality of the distribution of the estimated and baseline HR can be seen in Figure 4.4. The  $R^2$  values ranged between .935 to .980 for the estimated HR function, and between .771 to .785 for the baseline measurements. The agreement between the estimated and baseline HR was assessed using a Bland-Altman plot, as shown in Figure 4.5. In each Bland-Altman plot, an upwards trend is visible where larger values tend to show a greater difference.

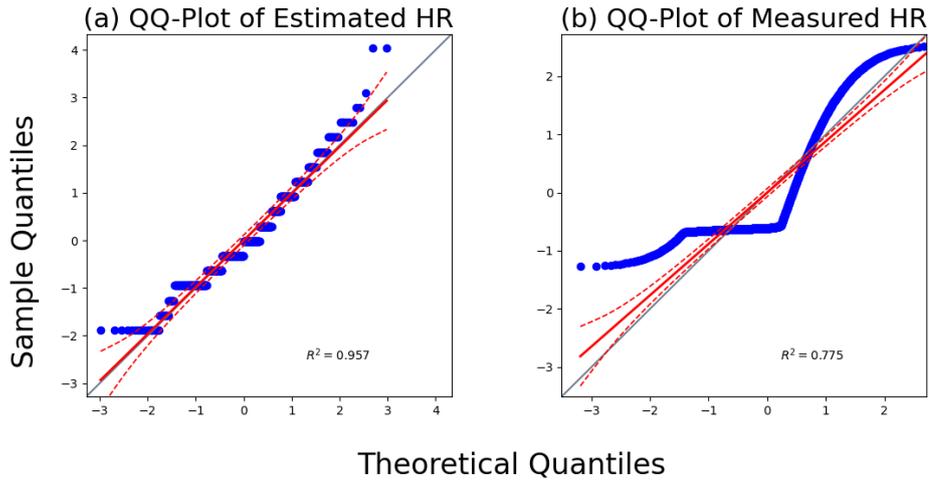


Figure 4.4: *QQ plots, showing the comparison between the distribution of estimated (a) and baseline (b) HR with the normal distribution.*

#### Bland-Altman Plots of Measured and Estimated HR

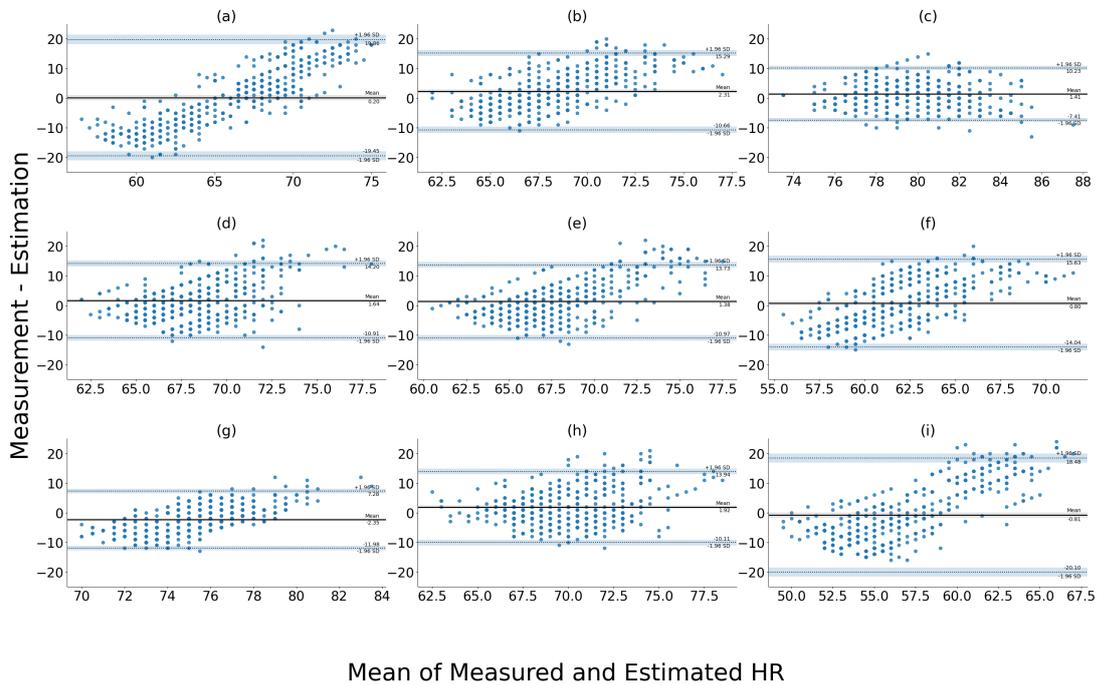


Figure 4.5: *Bland-Altman plots showing the agreement between the estimated and baseline HR. Note. The black line shows the  $M$  ranging from  $-2.35$  to  $2.31$ , and the limits of agreement are represented with blue lines, with a  $SD$  between  $-20.10$  to  $19.86$ .*

# Chapter 5

## Discussion

This thesis aimed to reconstruct and evaluate the model proposed by Li et al. [3]. The model’s algorithm incorporates EEMD for signal decomposition, PCA applied to specific IMFs, and FFT for the extraction of HR and RR [3]. To extend upon this, STFT was used instead of FFT, to extract time-resolved functions for HR and RR. Hyper-parameter tuning was done using grid search [20]. The model was evaluated using vibration and baseline data collected from a short-term controlled experiment. Due to faulty timestamp measurements, data from only 9 out of the 10 participants was valid for analysis. Nevertheless, this did not limit the evaluation of the model.

Overall, the model encountered significant challenges in processing noisy data and handling abrupt changes. Particularly movement noise consistently posed difficulties, a recurrent issue observed in Stage 2 of the experiment and supported by previous research [9]. The model’s average MAE, MPE, and MSE for HR estimation indicated an acceptable error margin, with the mean values of the estimated HR and baseline measurements being closely aligned, even when accounting for standard deviation. Nevertheless, cross-correlation analysis reveals a weak average correlation, and the variance of the baseline is notably higher than that of the model. This suggests that, despite the model appearing stable in comparison to the baseline at first glance, a closer inspection reveals that the resulted estimated HR function and the baseline measurements do not align well.

The model exhibits skewness close to zero (Table 4.4), suggesting a more or less symmetrical distribution. In contrast, the baseline shows positive skewness, indicating a longer right tail. The kurtosis values suggest that both distributions are close to normal. Further analysis using QQ plots revealed that the HR estimations are generally normally distributed. However, this distribution does not align with the baseline measurements, as indicated by the S-shaped curve in the QQ plot and the skewness values. In addition, the QQ plots highlight that the rounding and

averaging per second of the estimated HR introduced a grouping effect in the data, as evidenced by the step-like pattern in the left graph of Figure 4.4.

The Bland-Altman plots presented in Figure 4.5 reveal a gradual proportional bias between the estimated and baseline values. Although despite the visible distribution of the dots, a recognisable trend is still apparent, suggesting that while natural variability or noise may be present in the measurements, an overall systematic trend persists. The alignment of the dots into columns or clusters likely resulted from discretisation or rounding effects, caused by the averaging per second. Nonetheless, the overall trend appears to be upward.

A grid search, with parameter grids individually tailored for HR and RR estimation, was conducted for the STFT. This approach generally increased the model’s susceptibility to inconsistencies in the data. Particularly, because it seeks a general solution assuming a constant HR and RR, which is unrealistic given their natural variability. As visible in Plot (a) of Figure 4.3, the model struggled estimating the correct HR, when confronted with sharp transitions, such as the jump from the first to the second experimental stage. Additionally, the reliance on MSE as the primary evaluation metric presented challenges. Although the average error may appeared satisfactory, MSE did not adequately capture significant fluctuations in the data, potentially resulting in misleading conclusions regarding the model’s performance. An alternative approach could involve using other metrics such as the Root Mean Squared Error [9–11].

The parameters for HR estimation, identified via grid search, spanned a range of  $10$  to  $18$  samples per window, with an overlap of  $8$  to  $16$  samples. Thus, each window effectively only covered  $2$  new samples, necessitating the averaging of the extracted frequencies per second to accurately estimate HR. This approach introduced a vulnerability to outliers and false peaks, making the model highly dependent on the quality and characteristics of the vibration signal. The frequency ranges for HR estimation were initially set based on the assumption that the maximum HR was  $120$  bpm, with a minimum HR of  $48$  bpm, restricting the analysis to frequencies within  $0.8$  to  $2.0$  Hz. However, it was later discovered that the actual minimum HR was below  $48$  bpm, leading to incorrect results in certain cases. For instance, as shown in Plot (i) of Figure 4.3, the HR estimation in the first stage was inaccurately high because the actual HR was below  $48$  bpm.

For RR estimation, the frequency range of  $0.1$  to  $0.75$  Hz was analysed, as recommended by Li et al. [3]. Notably, no valid parameters were identified for RR estimation using grid search, resulting in the model’s inability to produce any results in this domain, highlighting another significant limitation in its performance. While this issue may stem from an incorrect parameter range defined for the grid

search, potential problems with geophone filtering, as suggested by Jia et al. [7], are unlikely. This is due to the fact, that the extracted IMFs do contain frequencies within the specified range, as evidenced by the spectrograms in Figure 4.1.

Compared to the results reported by Li et al., the accuracy of the reconstructed model is notably lower. Furthermore, the review of related works identified various approaches for extracting vital signs from vibration signals with superior performance. For example, the signal processing system "VitalMon" by Jia et al. [25] achieves a low HR and RR estimation error, even when two individuals share a bed. Similarly, the machine learning model "HeartQuake" by Park et al. [10] accurately detects all five ECG peaks with a low average error when participants are stationary. Additionally, it estimates clinically significant ECG metrics, such as the RR interval and QRS segment width, with low error rates. Looking towards future research, there is still potential in refining this approach. While Li et al. employed FFT to extract overall HR and RR from FPCs, our findings suggest that averaging the complete HR estimation from the STFT time series achieves nearly comparable accuracy. With further improvements in the extraction process, the STFT approach could provide more detailed temporal information. Additionally, improving the interval detection between beats or breaths, such as adaptive filtering [9], rather than relying solely on grid search, and exploring a potential hybrid approach incorporating non-black-box machine learning techniques, such as k-means clustering [6], could enhance the model's effectiveness.

In conclusion, the model did not deliver satisfactory results, achieving only partial success in HR estimation and failing to produce RR estimates. Improved interval detection methods, beyond the current grid search approach, are necessary to enhance the accuracy of the model. Future work should therefore explore the potential of integrating this approach with more robust heartbeat interval and respiration detection techniques, such as adaptive filtering, or machine learning approaches like k-means clustering.



# Appendix A

## Code

---

**Algorithm 11** Model based on the BaseEstimator from Scikit-learn [20]

---

```
from sklearn.base import BaseEstimator

class VitalSignExtractor(BaseEstimator) :
    def __init__(self, fs: int = 100,
                 win_len: int = None, noverlap: int = None,
                 nfft: int = None,
                 min_freq:float=0.8, max_freq:float=2.0):
        self.fs = fs
        self.min_freq = min_freq
        self.max_freq = max_freq
        self.win_len = win_len
        self.noverlap = noverlap
        self.nfft = nfft

    def fit(self, X, y) :
        self.is_fitted_ = True
        self.X = X
        self.y = y

        if not self.win_len :
            self.win_len = min(self.fs * 1/4, len(X))
        if self.noverlap " self.win_len :
            self.noverlap = self.win_len
        if not self.noverlap :
            self.noverlap = 0
        if not self.nfft or self.nfft " self.win_len :
            self.nfft = self.find_nfft(self.win_len)

        return self

    def predict(self, X) :
        # Feature extraction as described in main text
```

The VitalSignExtractor class, shown in Algorithm 11, is a custom Python implementation designed for extracting HR and RR from input data. This class inherits from the BaseEstimator class provided by the Scikit-learn [20] library, which facilitates integration with Scikit-learn’s machine learning framework.

# Appendix B

## Graphs

In the main text, we presented a QQ plot from a single representative participant. Below, additional QQ plots from a subset of participants to illustrate the variability in the results are included (Figure B.1 - B.4). Additionally, to compare baseline ECG and vibration data, 10 second excerpts are shown in Figures B.5 - B.8.

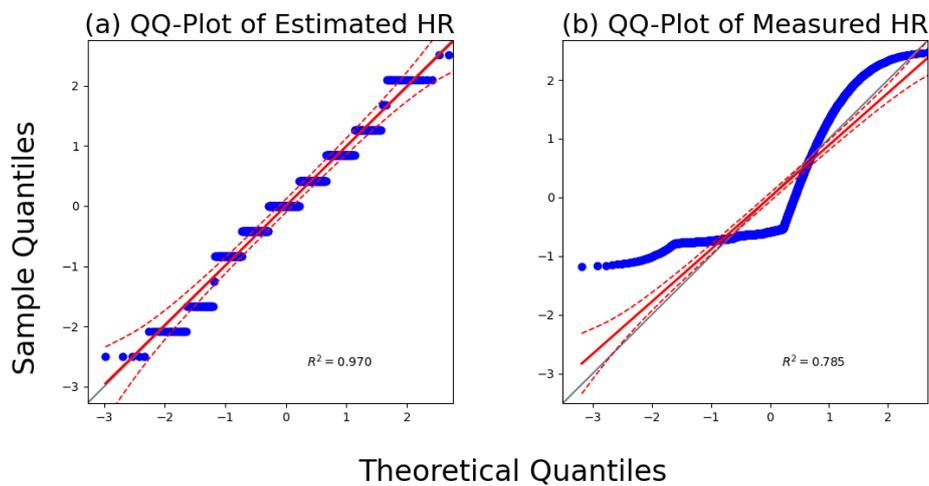


Figure B.1: *QQ plots, showing the comparison between the distribution of estimated (a) and baseline (b) HR with the normal distribution.*

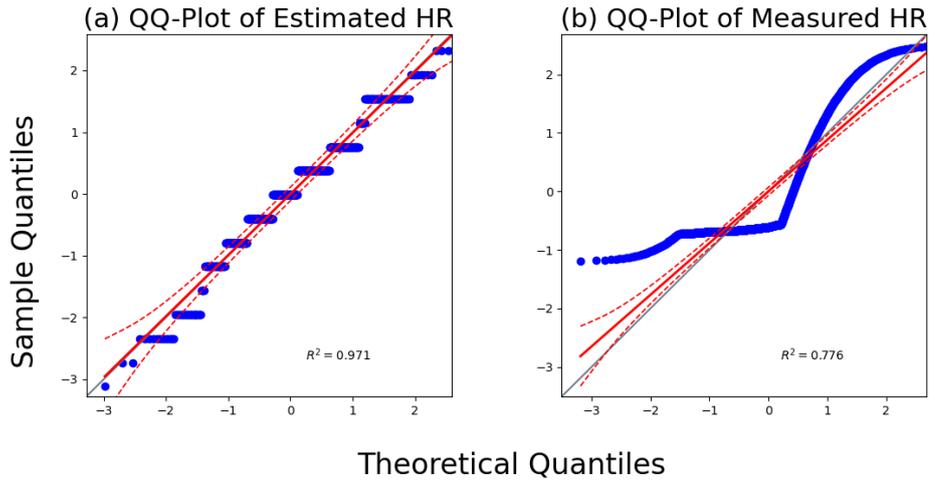


Figure B.2: *QQ plots, showing the comparison between the distribution of estimated (a) and baseline (b) HR with the normal distribution.*

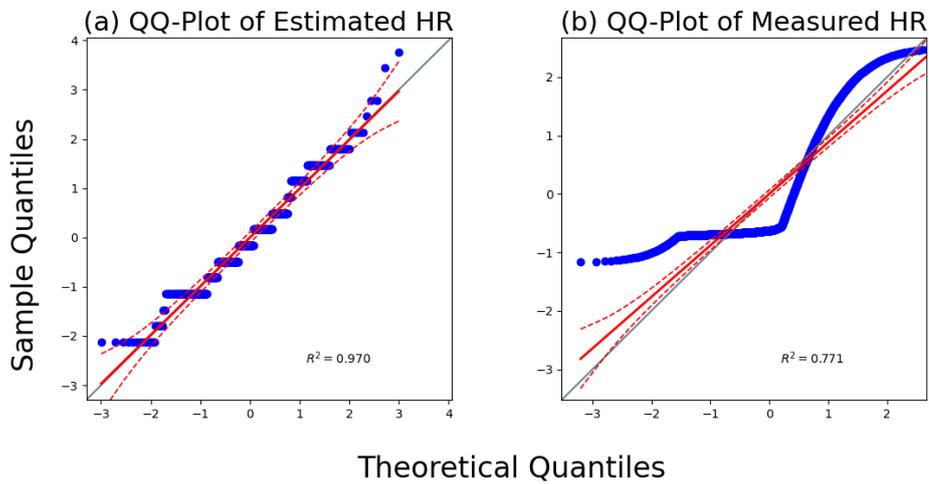


Figure B.3: *QQ plots, showing the comparison between the distribution of estimated (a) and baseline (b) HR with the normal distribution.*

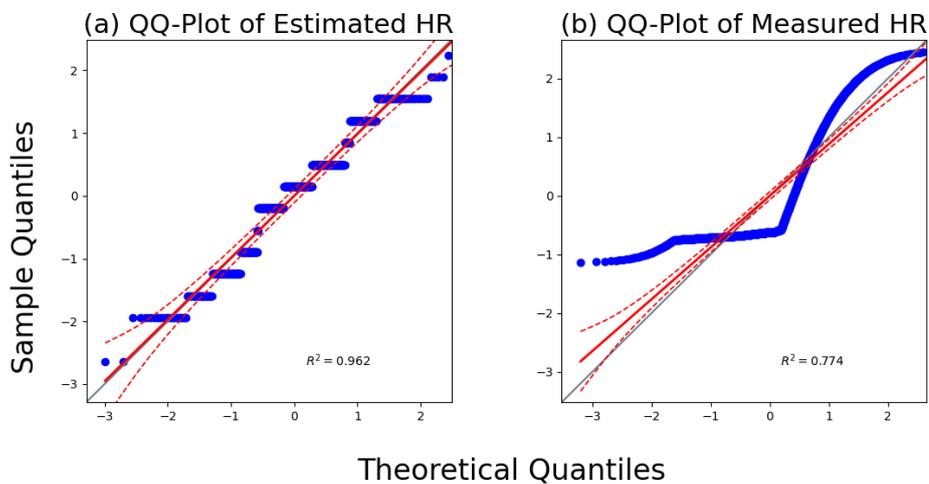


Figure B.4: *QQ plots, showing the comparison between the distribution of estimated (a) and baseline (b) HR with the normal distribution.*

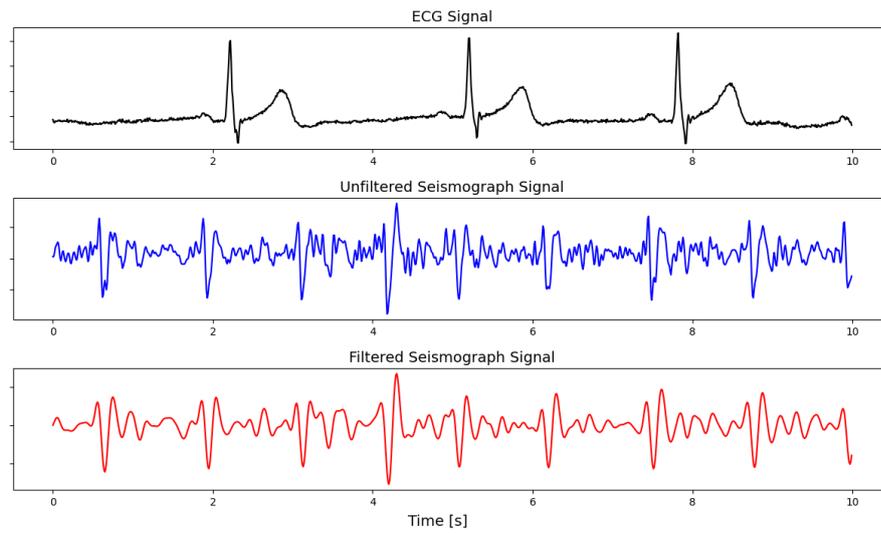


Figure B.5: A 10 second excerpt of the baseline ECG measurements (black) and the unfiltered (blue) and filtered (red) vibration signal

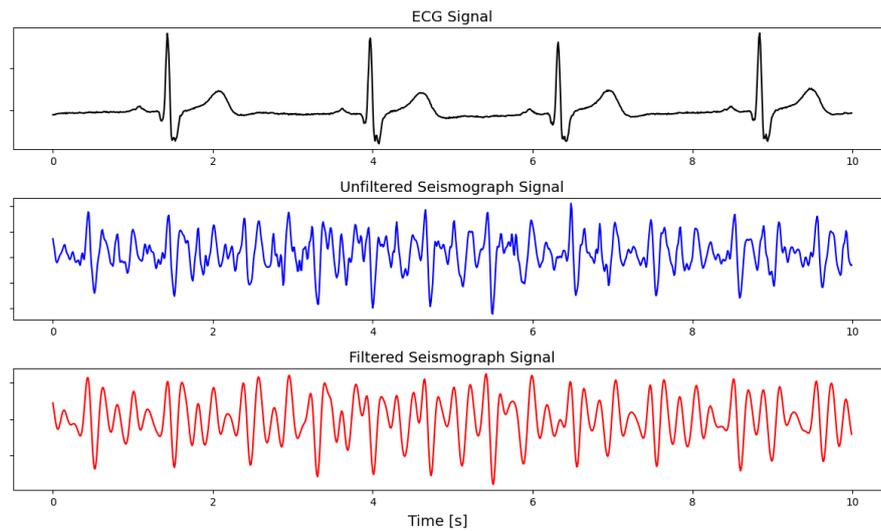


Figure B.6: A 10 second excerpt of the baseline ECG measurements (black) and the unfiltered (blue) and filtered (red) vibration signal

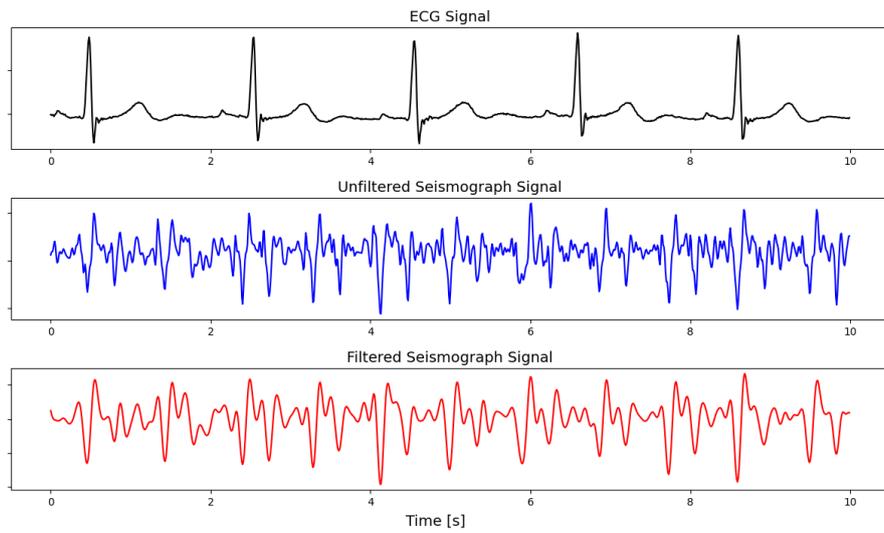


Figure B.7: A 10 second excerpt of the baseline ECG measurements (black) and the unfiltered (blue) and filtered (red) vibration signal

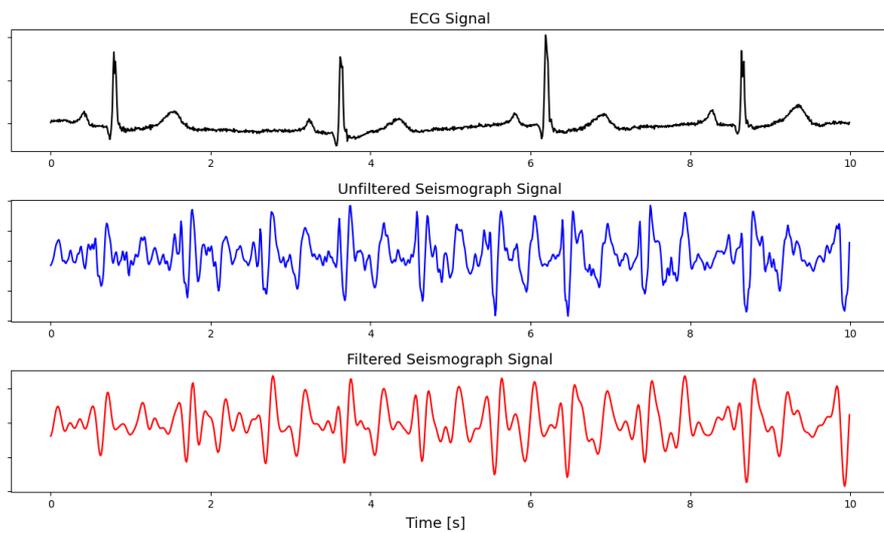


Figure B.8: A 10 second excerpt of the baseline ECG measurements (black) and the unfiltered (blue) and filtered (red) vibration signal

# Appendix C

## Statistical Results

Table C.1: *The duration in seconds of the collected data from the conducted study.*  
Note. Average per person is taken over the used data, which only covers the stages from the experiment.

Overall duration	7988.58
Used	4406.42
Average per person	489.6

Table C.2: *The MAE and MPE for the HR estimation for each participant, including an overall average.*

Participant	Mean Absolute Error	Mean Percentage Error
1	8.42	1.76
2	5.52	1.14
3	3.74	0.78
4	5.13	1.01
5	4.78	0.94
6	6.57	1.32
7	4.48	0.93
8	4.92	1.03
9	8.43	1.75
avg	5.78	1.18

# Appendix D

## Use of AI-based Tools

In this thesis, the Artificial Intelligence tool "Chat GPT" was used. All prompts used are listed below. The results from each prompt were used to get an intuitive understanding of a specific topic, or to rewrite existing text or code, according to specific criteria.

- Please rewrite the following into a paragraph in perfect formal British English for the "chapter name" chapter in a scientific paper. The paragraph should be 4 to 5 sentences with the first one as the introductory one. It must be clear, what the core meaning of the paragraph is. Please adhere to the Vancouver system citation style and use latex syntax: "bullet points of contents here"
- Please rewrite the following into paragraphs in perfect formal British English for the "chapter name" chapter in a scientific paper. Each paragraph should be 4 to 5 sentences with the first one as the introductory one. It must be clear, what the core meaning of the paragraph is. Please adhere to the Vancouver system citation style and use latex syntax: "bullet points of contents here"
- Please restructure and/or rewrite the following to better fit into a scientific paper written in formal British English in present tense. Please adhere to the Vancouver system citation style and use latex syntax: "text here"
- Please improve the flow and wording of the following paragraph. Use formal British English for a scientific paper. The paragraph should be 4 to 5 sentences with the first one as the introductory one. It must be clear, what the core meaning of the paragraph is. Please adhere to the Vancouver system citation style and use latex syntax: "text here"
- Please rewrite the following paragraph, so it is suited for a scientific paper in formal British English. The paragraph should be 4 to 5 sentences with the

first one as the introductory one. It must be clear, what the core meaning of the paragraph is. If it is better suited, you may split the paragraph into multiple paragraphs, which both must adhere to the previously stated rules. Please adhere to the Vancouver system citation style: "text here"

- Please shorten the following paragraph so it better suits a scientific paper in formal English. The paragraph should be 4 to 5 sentences with the first one as the introductory one. It must be clear, what the core meaning of the paragraph is. If it is better suited, you may split the paragraph into multiple paragraphs, which both must adhere to the previously stated rules. Please adhere to the Vancouver system citation style: "text here"
- Please shorten the following paragraph into "x" sentences in formal English for the "chapter name" chapter in a scientific paper. Please adhere to the Vancouver system citation style: text here"
- Please correct the grammar of the following text. Please use British English Vocabulary in present tense and adhere to the Vancouver system citation style: "text here"
- Restructure the following paragraphs to be coherent with each other. Each paragraph should be 4 to 5 sentences with the first one as the introductory one. Use British English vocabulary in present tense. It must be clear, what the core meaning of each paragraph is. Please adhere to the Vancouver system citation style. If it is better suited, you may split a paragraph into multiple paragraphs, which all must adhere to the previously stated rules: "paragraph here"
- Give me a longer list of synonyms for "word here", used in scientific papers.
- Rewriting method comments according to NumPy format: "Comment here"
- Give me an intuitive understanding of "topic here" and present some examples.
- Please write a method in python which does "xyz"

# Deceleration of Authorship

I herewith confirm that I wrote this thesis without external help and that I did not use any resources other than those indicated. I have clearly acknowledged all parts of the text where material from other sources has been used, either verbatim or paraphrased. I am aware that non-compliance with the above statement may lead to withdrawal of the academic title granted on the basis of this bachelor's thesis by the Senate, according to the law governing the University of Bern. If I use artificial intelligence as an aid, I must declare all elements that derive from it. I must list the name of the technology as well as the search terms "prompts" that I have used. I am aware that otherwise the thesis will receive the grade 1.

30.08.2024

A handwritten signature in black ink, consisting of a large, stylized 'R' followed by a series of loops and a long horizontal stroke.

# Bibliography

- [1] Peter Kowallik and Hubert Wirtz. Method and device for sleep monitoring. Publisher: United States Patent and Trademark Office Type: Patent.
- [2] Subhas Chandra Mukhopadhyay. Wearable sensors for human activity monitoring: A review. 15(3):1321–1330.
- [3] Fangyu Li, Maria Valero, Jose Clemente, Zion Tse, and Wenzhan Song. Smart sleep monitoring system via passively sensing human vibration signals. 21(13):14466–14473.
- [4] Michaela Sidikova, Radek Martinek, Aleksandra Kawala-Sterniuk, Martina Ladrova, Rene Jaros, Lukas Danys, and Petr Simonik. Vital sign monitoring in car seats based on electrocardiography, ballistocardiography and seismocardiography: A review. 20(19):5699.
- [5] Ibrahim Sadek, Jit Biswas, and Bessam Abdulrazak. Ballistocardiogram signal processing: a review. 7(1):10.
- [6] Omar Y. López-Rico and Roberto G. Ramírez-Chavarría. Smart seismocardiography: A machine learning approach for automatic data processing. In *The 8th International Electronic Conference on Sensors and Applications*, page 24. MDPI.
- [7] Zhenhua Jia, Musaab Alaziz, Xiang Chi, Richard E. Howard, Yanyong Zhang, Pei Zhang, Wade Trappe, Anand Sivasubramaniam, and Ning An. HB-phone: A bed-mounted geophone-based heartbeat monitoring system. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12. IEEE.
- [8] S.A. Raspberry Shake. Raspberry shake global technical specifications document. Published: [\texttt{https://shop.raspberrypi.org/}](https://shop.raspberrypi.org/).
- [9] Mojtaba Jafari Tadi, Eero Lehtonen, Tero Huranenen, Juho Koskinen, Jonas Eriksson, Mikko Pänkäälä, Mika Teräs, and Tero Koivisto. A real-time ap-

- proach for heart rate monitoring using a hilbert transform in seismocardiograms. 37(11):1885–1909.
- [10] Jaeyeon Park, Hyeon Cho, Rajesh Krishna Balan, and JeongGil Ko. HeartQuake: Accurate low-cost non-invasive ECG monitoring using bed-mounted geophones. 4(3):1–28.
  - [11] Yuhang Chen, WenChang Xu, Wenliang Zhu, Gang Ma, Xiaohe Chen, and Lirong Wang. Beat-to-beat heart rate detection based on seismocardiogram using BiLSTM network. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1503–1507. IEEE.
  - [12] JaeYeon Park, Hyeon Cho, Wonjun Hwang, Rajesh Krishna Balan, and Jeong-Gil Ko. Deep ECG wave estimation model with seismograph sensor (poster). In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 568–569. ACM.
  - [13] Michael Chan, Venu G. Ganti, and Omer T. Inan. Respiratory rate estimation using u-net-based cascaded framework from electrocardiogram and seismocardiogram signals. 26(6):2481–2492.
  - [14] Thomas P. Quinn, Stephan Jacobs, Manisha Senadeera, Vuong Le, and Simon Coghlan. The three ghosts of medical AI: Can the black-box present deliver? 124:102158.
  - [15] Gregor Stiglic, Primoz Kocbek, Nino Fijacko, Marinka Zitnik, Katrien Verbert, and Leona Cilar. Interpretability of machine learning-based prediction models in healthcare. 10(5):e1379. Publisher: Wiley Periodicals LLC.
  - [16] svgsilh.com. Silhouette image of a person walking. <https://svgsilh.com/image/310276.html>, 2024. Accessed: 2024-08-30.
  - [17] Ghufran Shafiq, Sivanagaraja Tatinati, Wei Tech Ang, and Kalyana C. Veluvolu. Automatic identification of systolic time intervals in seismocardiogram. 6(1):37524.
  - [18] The pandas development team and contributors. pandas: Powerful data structures for data analysis in python.
  - [19] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, and others. SciPy 1.0: Fundamental algorithms for scientific computing in python. 17(3):261–272.

- [20] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. 12:2825–2830.
- [21] Lubomir M. Lapka. PyEMD: Python implementation of empirical mode decomposition.
- [22] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, and others. Array programming with NumPy. 585(7825):357–362. Publisher: Nature Publishing Group Version Number: 1.26.0.
- [23] Raphael Vallat. Pingouin: Statistics in python.
- [24] John D. Hunter. Matplotlib: A 2d graphics environment.
- [25] Zhenhua Jia, Amelie Bonde, Sugang Li, Chenren Xu, Jingxian Wang, Yanyong Zhang, Richard E. Howard, and Pei Zhang. Monitoring a person’s heart rate and respiratory rate on a shared bed using geophones. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, pages 1–14. ACM.
- [26] Jose Clemente, Fangyu Li, Maria Valero, and WenZhan Song. Demo: Contactless device for monitoring on-bed activities and vital signs. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 472–474. IEEE.