Contactless Estimation of Breathing Rate Using UWB Radar

A.K. Isuru Udayanga W. Gunasekara

A thesis submitted in partial fulfillment of the requirements for the Master's degree in Electrical Engineering and Computer Science

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa

Supervisor: Prof. Miodrag Bolic

Abstract

Contactless breathing estimation using radars has been explored since the 1960s and an accurate system with the ability to continuously monitor the health of non-critical patients without obstructing their day to day lives could significantly improve their well being. The current state of the art in this area does not have the accuracy required to work in a real-world environment and many of the existing methods have been tested only under very controlled situations. Low performance of breathing estimation algorithms under different scenarios inspired us to improve breathing estimation algorithms and develop a system for automated analysis of large number of algorithms against data from the reference sensors. A novel accurate breathing rate estimation method and a system to use multiple algorithms on the same set of data in real-time and identify the best performing algorithm dynamically to report breathing rate have been proposed in this thesis. In addition, automated data-collection and processing frameworks were developed to collect a large amount of data and process them and generate reports automatically. The proposed system has been tested under multiple test-cases involving multiple subjects and the accuracy of both new and existing algorithms have been evaluated by comparing the results with reference data collected using a respiration belt. The mean absolute error rate of breathing rate estimation after conducting experiments for a total of 9 subjects was found to be 0.79 breaths per minute for the novel CEEMD based method presented in this thesis. The mean absolute error rate after applying the scoring algorithm to select the best performing algorithm is 0.78 breaths/minute.

Acknowledgement

I would like to thank my supervisor, Dr. Miodrag Bolic for the support he provided me for this work. He was always helpful during any time I ran into issues or had any question and I would also like to thank him for believing in me and giving me the opportunity carry out this research work under his supervision. I would also like to thank Dr. Sreeraman Rajan from Carleton University for giving me advice and helping me in this research. Both him and Dr. Bolic helped out in numerous situations such as, reserving and arranging laboratory environments for data collection/testing, getting ethics approval for tests, helping out in transporting equipment etc. in addition to helping out in conventional methods such as help solve problems on paper.

I would also like to thank Zachary Baird, Dr. Munir Tarar, Dr. Isar Nejadgholi, Britto Martial, Hamza Qassoud, Aly Khedr, Samuel Hadwen, Xander May, Anwar Fallatah and Gangesh Beri who helped out in numerous ways such as helping out in conducting experiments, developing algorithms/code, working on commercializing this research, and in many other ways. Dr. Bolic, Dr. Tarar, and Dr. Rajan, always believed in the potential of this research work and worked tirelessly to apply this research in real world situations to help elderly people in elder care facilities.

Finally I would like to express my sincere gratitude to my parents and my brother who always believed in me and helped me achieve all that I have achieved in life. My parents always put helping me in front of all other things in their life and their selfless commitment to help me have a better life truly amazes me.

Contents

List of Figures

List of Tables

Acronyms

CEEMD	Complete Ensemble Emprical Mode Decomposition
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
EMD	Empirical Mode Decomposition
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FFT	Fast Fourier Transform
IC	Industry Canada
IMF	Intrinsic Mode Function
NMD	Non-linear Mode Decomposition
UWB	Ultra-wide band

Chapter 1

Introduction

1.1 Motivation

Continuous, real-time and non-invasive sensing of vital signs have the potential to improve a person's well being. This is especially true if the person is elderly and/or have medical conditions that requires them to be monitored continuously. Making a contactless system robust enough to be able to do this task with minimal human or physical intervention would have the ability to ease the day to day life of those who are being monitored.

Radar technology has been explored in estimating respiration since the 1960s [1]. However, it has not been used in a commercial application successfully and most research work are constrained to testing radars under laboratory environments and many studies are limited to only a few experiments conducted under very controlled environments. One of the main focuses during this thesis work was to develop a system that can be used in real-world situations incorporating existing and novel algorithms with the ability to collect, process and send respiration information to caregivers automatically.

Figure 1.1 shows a summary of the motivation behind this research work and its outcomes. As seen on Figure 1.1, the major issues with existing algorithms are their lack of accuracy in realistic environments which limits them from being used in such environments. It was also identified during this work that a single algorithm can not provide accurate results for breathing rate estimation in all cases as most of the algorithms perform well in very specific situations. Therefore a system to analyze the results from multiple algorithms over time (Section 3.2.4) was developed and this system selects the best performing algorithm dynamically to

report the breathing rate. In addition to existing algorithms, a novel ensemble empirical mode decomposition based algorithm was developed to estimate breathing rate and this algorithm is also included in the scoring system that selects the best performing algorithm. Most research work does data processing separately after collection and since this is not suitable for a system that could be used in a real-world situation to monitor humans, the ability to collect and process the data simultaneously was built in to the final system developed using Python™ and Matlab®.

A very large number of test-cases are required to test a system under multiple test cases in real-world situations. If the analysis for the performance of a system using multiple algorithms on multiple test-cases was done manually, the researcher(s) would have to spend a lot of time just collecting, labelling and processing the data. To avoid this, an automated system was developed to collect, label and save data both from radar sensors (Compatible with both radar sensors used in this thesis and compatibility for other sensors can be added easily due to the modular nature of the application) and a reference sensor such as a respiration monitoring belt. Two separate applications (A Matlab® based system to generate 'pdf' and 'csv' reports and the Python™ based real-time application described above was modified to give it the ability to analyze saved data files and generate 'pdf' and 'csv' reports instead of running in real-time mode) were developed which are able to load the saved radar data and reference data files and generate reports for multiple test-cases automatically.

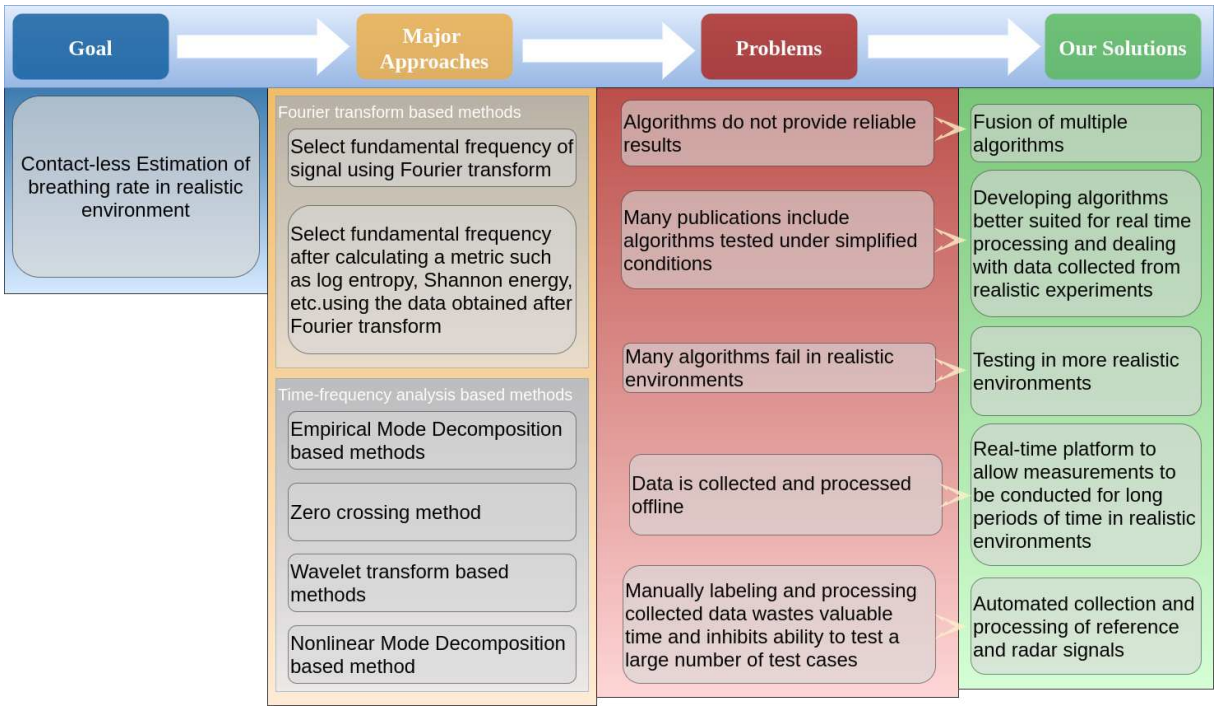


Figure 1.1: Motivation behind research and outcomes

1.2 Contributions

The following research and development based contributions were produced during this thesis work:

1. Research based contributions

    - A novel ensemble empirical mode decomposition based algorithm (described in Section 3.2.2) for estimating breathing rate.
    - A scoring algorithm capable of selecting the best performing algorithm dynamically. (Section 3.2.4)
    - A research paper [2] was published in the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS) in collaboration with Z. Baird, M. Bolic and S. Rajan, and two more publications are currently under development based on the work presented in this thesis.

2. Development based contributions

- A commercially deploy-able system with a Python™ based application (described in Section 3.6) capable of collecting data from a radar (compatible with two different radar models as described in Section 2.1.1), processing the data using multiple algorithms, estimating breathing rate and sending the results to a local GUI and a cloud server.
- An automatic data collection application with the ability to collect data from multiple sources such as radar, reference from a breathing belt, camera, was developed to collect and label data pertaining to multiple test-cases without having to manually collect and label data files. The collected data is also automatically synchronized with each other so that during processing, the reference data can be compared to the estimations easily. (Section 3.3)
- Two automatic data processing applications capable of generating reports about the performance of algorithms in estimating breathing rate with the ability to run a large amount of data files and generate multiple reports at once. (Sections 3.5 and 3.6)

Chapter 2

Background

Breathing rate estimation using radar sensors usually involves the following steps:

- Pre-processing the data to remove noise/clutter.
- Identifying the location of the subject.
- Estimate breathing rate.

This chapter provides an introduction about how the radar systems used in this thesis works (Section 2.1.1) and a background about the existing methods/algorithms used for physiological measurements using radars.

Section 2.2 introduces how noise/clutter is removed in radar signals and how the location of the subject can be estimated and Section 2.3 shows several methods to estimate breathing rate using the extracted signal from the subject location.

2.1 Radar technology in physiological measurements

Radar technology has been used to sense physiologic movement since the 1960s [1]. However, limitations and cost of hardware and lack of proper robust algorithms limits the use of such research in mainstream commercial applications. Recent advances in microcircuit fabrication and wireless technology is enabling the cost of radar sensors to reduce and advances in signal processing and machine learning techniques would enable radar technology to be used for reliable respiration monitoring in the near future.

Respiration monitoring using radar technology has been explored about mainly using two types of radar sensors:

- Continuous wave (CW) radar
- Pulse radar

In this thesis, the use of Pulse radar is explored for respiration measurement and continuous wave radars are not discussed in this work.

2.1.1 Pulse Radar

A pulse radar operates by emitting a short pulse and then receiving the echo signal in the silent period. Unlike a continuous wave radar, the transmitter is turned off before the measurement is finished. The transmitted pulse consists of very short transmission pulses (typically in durations of few ns) and comparatively large pauses between each pulse transmission (typically in durations of few ms) which are used to receive the reflected pulses [3]. The small time duration of the transmitted pulse and its shape means that the transmitted signal has a very wide band-width in frequency domain and hence these types of radars fall into the Ultra-Wide-Band (UWB) radar category. The FCC defines UWB signals as those that occupy a 10-dB bandwidth of greater than 500 MHz band-width or larger than 20% fractional bandwidth over the unlicensed frequency range of 3.1-10.6 GHz, where fractional bandwidth is defined as shown in equation 2.1

$$\text{Fractional Bandwidth} = \frac{2(f_H - f_L)}{f_H + f_L} \tag{2.1}$$

where f_H and f_L are the upper and lower frequency limits respectively, of the 10-dB band-width [4]. The FCC also requires that the power emission levels of the UWB signals within the UWB spectrum of 3.1-10.6GHz must be low to avoid interference with other technologies operating in the same frequency bands and requires the maximum allowed power spectral density (PSD) not to exceed -41.25 dbm/MHz [5]. This limited emitting power presents a serious challenge to UWB systems as they are operating in a band of operation with other devices such as Wireless Fidelity (WiFi) [6] and these RF systems sharing the same bandwidth usually have a much higher transmitting power. This low RF emitting power requirement limits UWB systems

to work only within short ranges, making it suitable to employ minature CMOS RFICs whose RF power capability and dc power consumption are relatively small [5].

Compared to CW based systems, UWB impulse systems have many advantages such as:

1. **Fine resolution and long range:** A UWB system by it's definition enables fine range-resolution because the range resolution is inversely proportional to the bandwidth. The presence of low and high frequencies enable long range as compared to a CW system containing only high frequencies. While similar range-resolution and range can be achieved by a CW based system operating in similar bandwidth and frequencies, it is not practical or economical to realize an extremely wide-band CW system [5].

2. **High multi-path resolution and low interference with other existing signals:** The transmit energy spectral density of UWB impulse systems are typically much lower than CW systems since the total energy is spread over a very wide frequency range. Hence a UWB impuse radar is unlikely to interfere with any other existing signal. The impulse signals are also very short in time-domain and hence the multi-path signals have a very short time-window of opportunity to collide with line of sight signals and less likely to cause signal degradation.

3. **Simple and low-cost architecture:** UWB impulse systems can be implemented with a simpler architecture than wide-band CW systems [5].

The UWB impulse radars used in this research also has an added advantage of using coherent signal processing which enables the systems to sum the returns of several pulses as systems with coherent processing maintains the phase information of each pulse. This gives the radars used in this research the added advantages of increased operating range and SNR [7].

UWB impulse radars work by transmitting an impulse (The word impulse is loosely used. The transmitted signal is not truly an impulse but a pulse with very narrow pulse width in time domain) signal modulated by a carrier frequency. Gaussian-like impulse, doublet pulse and monocycle pulse are typically used in UWB pulse radar systems [5].

A Gaussian impulse can be expressed as shown in Equation 2.2.

$$p(t) = a \cdot \exp\left(- \frac{(t - \frac{T_p}{2})^2}{2\sigma_p^2} \right) \tag{2.2}$$

Where a is the maximum amplitude and T_p and σ_p^2 are the duration and the variance of the Gaussian pulse respectively.

Such a pulse is then modulated using a high frequency carrier signal and this process is illustrated in Figure 2.1 and Figure 2.2 shows a block diagram of a typical UWB pulse transmitter. It consists of a pulse generator and a digitally controlled modulator circuit that controls the timing or polarization of the transmitted pulse signal (Out of the two radar modules used in this research, only the X4M03 module by Xethru® has the ability to transmit bi-phase pulses, *i.e.* "Data In" signal alternates between 1 and 0 for transmitting pulses with 180 degree phase shift between each other). A local oscillator determines the pulse repetition frequency (PRF) of the UWB impuse signal and the Pulse Generator produces a pulse signal with a desired waveform such as a Gaussian pulse.

The transmitted signal in time domain after these steps for the ith frame is given by Equation 2.3.[2]

$$s_i(t) = p(t - iT_f) \cdot cos(2\pi f_c(t - iT_f)) \tag{2.3}$$

where,	f_c	Carrier frequency
	f_p	Pulse repetition frequency
	T_f	Duration of frame $T_f = \frac{1}{f_p}$

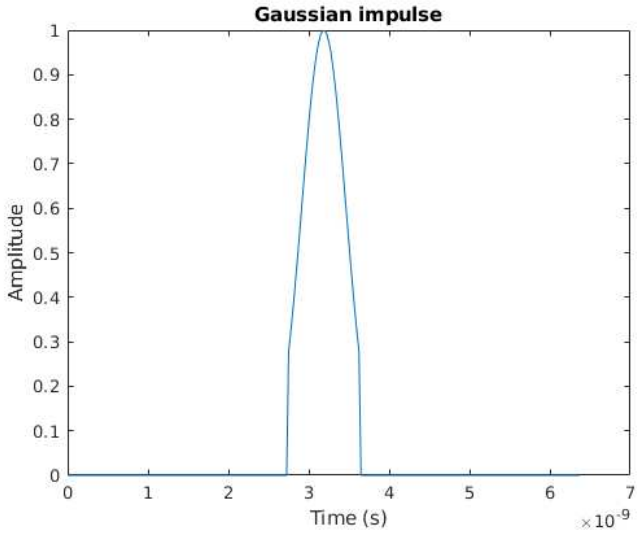
Let us define a local frame time variable t_f as $t = t_f + iT_f$ with $t_f \in [0, T_f]$ to simplify Equation 2.3. The transmitted signal for frame i can now be written as;

$$s_i(t_f) = p(t_f) \cdot cos(2\pi f_c t_f) \tag{2.4}$$

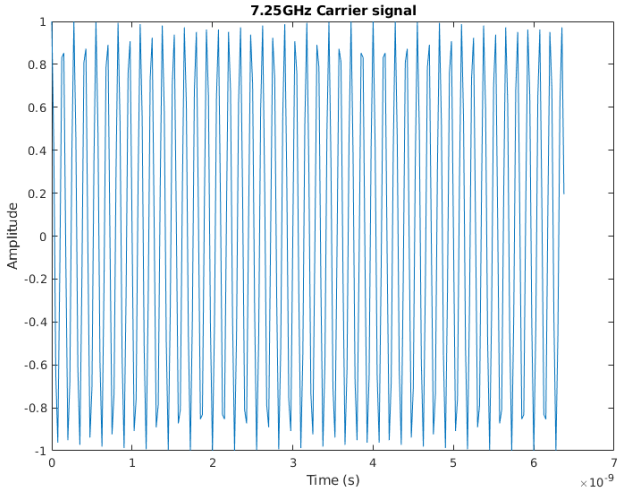
Figure 2.3 shows the block diagram of a typical UWB receiver circuit which consists of a low-noise amplifier (LNA), a correlation circuit and a template pulse generator. The oscillator drives the pulse generator and determines the pulse repetition frequency (PRF) of the UWB impulse system. After passing the LNA the received pulse is coherently correlated with the template pulse generated by the pulse generator through the correlator and the input pulse is then converted into a baseband signal in a single stage just using the correlator which reduces the

[1] Source: Design of CMOS RFIC Ultra-Wideband Impulse Transmitters and Receivers. Cam Nguyen, Meng Miao

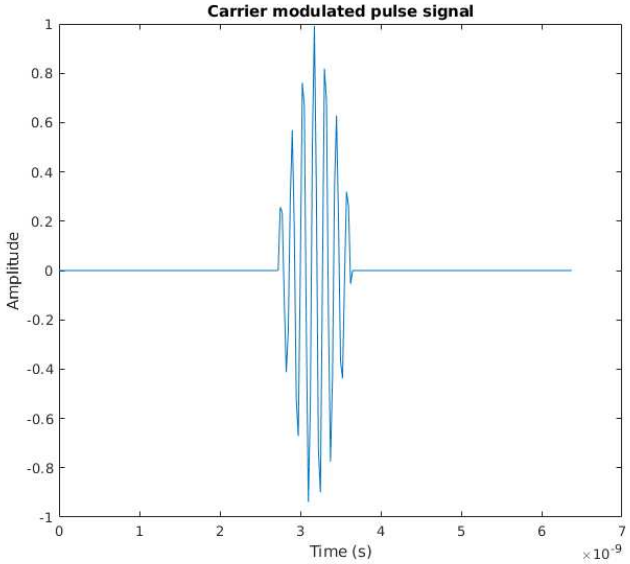
[2] Any amplification stages before transmission are ignored when expressing this equation to maintain clarity.



(a) Gaussian pulse signal (36 point impulse padded with 110 zeroes on each side. $\alpha = 1.6$, α is inversely proportional to the standard deviation of the Gaussian distribution)



(b) 7.25GHz carrier signal



(c) Modulated pulse signal

Figure 2.1: Process of modulating a Gaussian pulse with a carrier signal (Example signals created using Matlab)

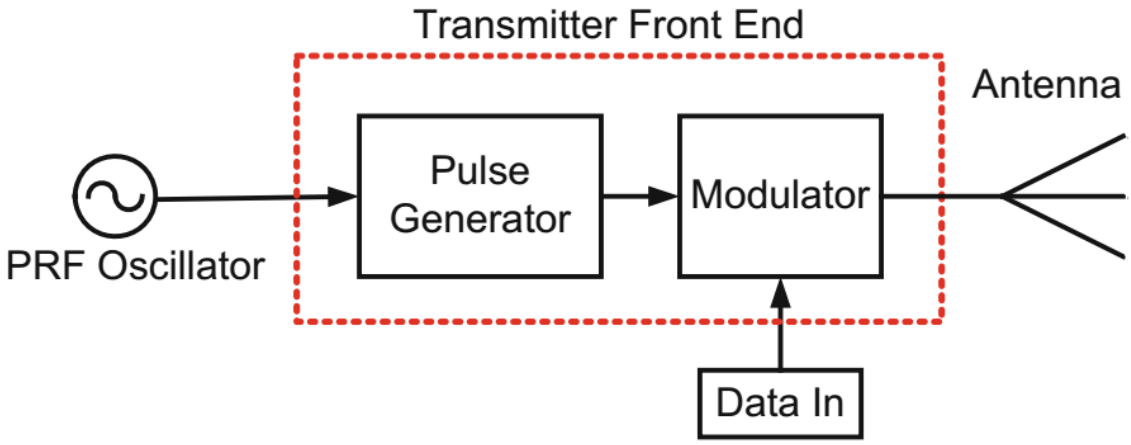


Figure 2.2: Block diagram of a typical pulse UWB transmitter[1]

system complexity. The time domain and frequency domain representation of such a signal after the baseband conversion can be found in Figure 2.4.

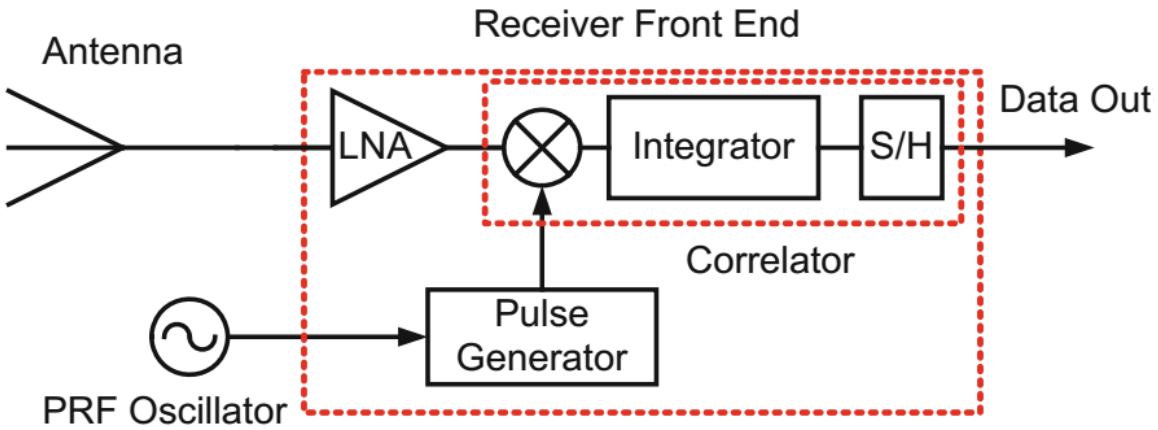


Figure 2.3: Block diagram of a typical pulse UWB receiver[3]

The returned signal from a single target for the ith frame is given by Equation 2.5 [8].

$$\begin{aligned} r_i(t_f) &= G_{Rx} \cdot L(t_f) \cdot s_i(t_f - \tau(t)) \\ &= G_{Rx} \cdot L(t_f) \cdot p(t_f - \tau(t)) \cdot \cos(2\pi f_c(t_f - \tau(t))) \end{aligned} \tag{2.5}$$

[3]Source: Design of CMOS RFIC Ultra-Wideband Impulse Transmitters and Receivers. Cam Nguyen, Meng Miao

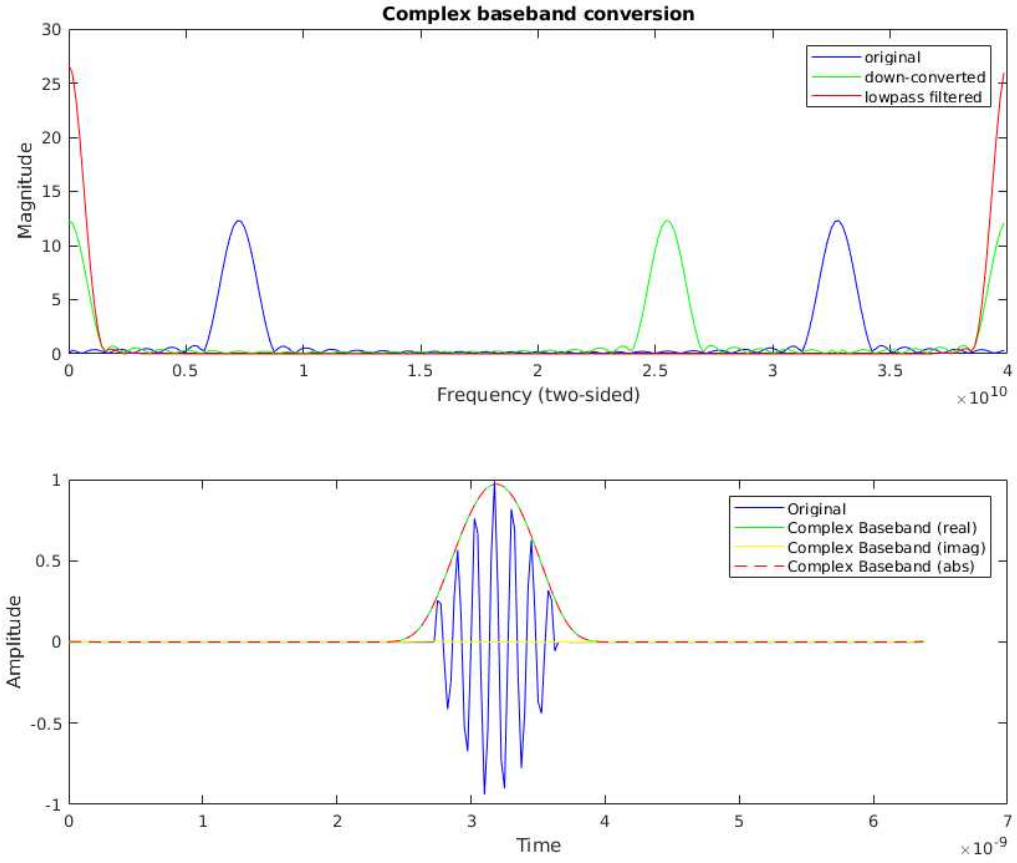


Figure 2.4: Time and frequency domain representation of a complex baseband signal after down-conversion

where,	G_{Rx}	Receiver amplifier (LNA) gain.
	$\tau(t)$	Propagation delay equal to the radar pulse round-trip time
	$L(t)$	Free space loss

and,

$$\tau(t) = \frac{2 \cdot R(t)}{c} = \frac{2 \cdot R(t_f + iT_f)}{c} \tag{2.6}$$

where,	$R(t)$	Range to target
	c	Speed of light

And the free space loss $L(t)$ is given by $L(t) = \sqrt{\frac{P_r}{P_t}}$ where P_t and P_r are the transmitted power and the received power respectively.

The received signal power P_r from a target at the LNA input is given by equation 2.7 [9]

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R_t^2 R_r^2} \tag{2.7}$$

where,

G_t	Gain of transmitting antenna
G_r	Gain of receiving antenna
λ	Wavelength of the radar carrier signal
σ	Target equivalent radar cross section (RCS) (*i.e.* radar cross section of target)
R_t	Range from transmitting antenna to the target
R_r	Range from receiving antenna to the target

As both antennas used in this thesis (X4A02 Antenna for for the X4M03 module and Sinuous antenna module for Ancho™ radar module as described below) are in monostatic configuration [10, 11], R_t and R_r are nearly identical, hence let's replace R_t and R_r with R, and similarly G_tG_r can be replaced with G^2G_{Rx} where G_{Rx} is the receiver LNA gain as described above. Since the targets are moving R will be a time-dependent function $R(t)$. Equation 2.7 can be modified as;

$$P_r(t) = \frac{P_t G^2 \lambda^2 \sigma G_{Rx}}{(4\pi)^3 R(t)^4} \tag{2.8}$$

Hence, the free space loss $L(t)$ can be rewritten as;

$$L(t) = \sqrt{\frac{G^2 \lambda^2 \sigma}{(4\pi)^3 R(t)^4}} \tag{2.9}$$

Plugging all of this into Equation 2.5 gives;

$$\begin{aligned} r_i(t_f) = G_{Rx} \cdot \sqrt{\frac{G^2 \lambda^2 \sigma}{(4\pi)^3 R(t_f)^4}} \cdot p\left(t_f - \frac{2 \cdot R(t_f + iT_f)}{c}\right) \\ \cdot cos\left[2\pi f_c \left(tf - \frac{2 \cdot R(t_f + iT_f)}{c}\right)\right] \end{aligned} \tag{2.10}$$

The signal is sampled at $(t_s + nT_s)$ where t_s is the frame offset time defining the minimum range of the radar, T_s is the sampling period (A conventional ADC is much slower to achieve the required sampling rate for a radar, therefore, both radar modules used in this thesis uses multiple parallel samplers to achieve high sampling rate) and f_s is the sampling frequency (*i.e.* $T_s = \frac{1}{f_s}$) and $n = 0, 1, 2, \ldots N-1$ where N is the number of samples to be recorded for each frame.

$$r_i[n] = r_i(t_s + nT_s) = G_{Rx} \cdot L(t_s + nT_s) \cdot p\left(t_s + nT_s - \frac{2 \cdot R(t_s + nT_s + iT_f)}{c}\right) \cdot \cos\left[2\pi f_c\left(t_s + nT_s - \frac{2 \cdot R(t_s + nT_s + iT_f)}{c}\right)\right] \tag{2.11}$$

Next the complex base-band conversion mentioned above is performed by multiplying $r_i[n]$ with a complex phasor with the same frequency as the radar carrier frequency. The base-band signal $r_{i,bb}[n]$ is; [8]

$$r_{i,bb}[n] = r_i[n] \cdot \exp(-j2\pi f_c(iT_f + nT_s)) \tag{2.12}$$

The radars used in this thesis also has a feature where each data frame is recorded by sending multiple frames repeatedly and summing up the received result over time. As explained before this is possible because of coherent signal processing and this improves the signal to noise ratio.

An important concept in explaining UWB pulse radar signals is the concept of slow-time and fast-time. Slow time is the time between each pulse(If only one pulse is used without pulse integration as described in the previous paragraph) and the fast time is the time between each sampling instance on the receiver. Consequently, fast-time corresponds to the distance of an object from the radar. Some more details about slow-time and fast-time can be found at the end of this section in Figures 2.6 and 2.7.

The following two models of pulsed radars were used in this research:

- FlatEarth Ancho™ Radar module[4]
- Xethru®X4M03 custom sensor development platform [5]

The FlatEarth Ancho module consists of a Xethru®X2 radar SOC, Sinuous antenna, and a Beaglebone™Black single board computer. The Xethru X2 chip directly connects to the SPI interface of a Beaglebone™Black SBC and a USB interface is provided to get data to a computer connected to the SBC. The Xethru®X2 SOC uses a concept called strobed sampling in which the back-scattered electromagnetic waves are sampled after a given time offset which represents the time-of-flight of the signal relative to the time of pulse transmission [12]. Xethru®X2 SOCs use 256 parallel samplers where each sampler is triggered at a different time offset and this increases

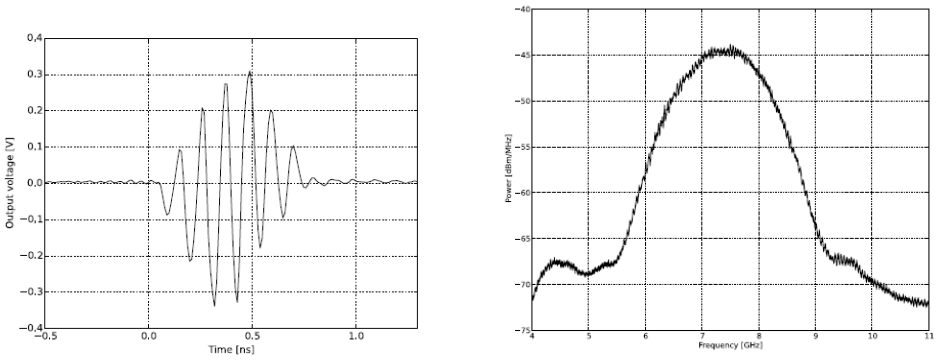
[4] https://store.flatearthinc.com/products/ancho-radar-development-kit-sinuous
[5] https://www.xethru.com/xethru-development-platform.html

the spatial resolution of the radar. Figure 2.5 shows a pulse generated by the X2 SOC in time and frequency domain. The sampling rate of the radar can be calculated using equation 2.13

$$F_s = \frac{N_{samplers}}{t_{delayline}} \tag{2.13}$$

Where $N_{samplers}$ stand for the number of parallel samplers (256 for the X2 SOC) and t is the total delay time of the sampler's line [13]. With $t \approx 6.5$ ns, the sampling rate of the radar (Intra-frame sampling rate) can be calculated as $F_s \approx 40$ GS/s.

The pulses are generated using an internal pulse generator and the pulse shape and bandwidth can changed by selecting one of 11 pulse generators available in the X2 SOC. The pulse shape used in this research is shown in Figure 2.5[6]. This pulse shape was used as it matches the recommended operating frequency range of the sinuous antenna used with the Xethru® X2 SOC [11]



(a) Time domain signal (b) Frequency domain signal

Figure 2.5: Pulse generator output of FlatEarth Ancho™radar

The radar SOC (Xethru X2) transmits pulses at the pulse repetition frequency (PRF) (100Mhz in this case) and the sweep controller is configured for bi-directional sweeping, *i.e.* one sweep will range from DACMin(minimum DAC value) to DACMax(maximum DAC value), incremented by the DACStep(increment from one DAC value to the next). Therefore after receiving the radar frames from the radar SOC the data has to be normalized in order to get similar data (amplitude wise) irrespective of the radar settings. The normalization is done according to equation 2.14 to

[6]A different pulse shape was used for a few experiments but for the sake of simplicity, it is assumed that all experiments used the same pulse generator

get a DAC count between 0 and 8191 and equation 2.15 can be used to convert the value into a voltage [12]

$$\text{NormalDACCount} = \text{DACCount} \cdot \left(\frac{\text{DACStep}}{\text{PulsesPerStep} \cdot \text{Iterations}} \right) + \text{DACMin} \tag{2.14}$$

$$\text{NormalVolt} = \text{NormalDACCount} \cdot \left(\frac{1.04}{8191} \right) \tag{2.15}$$

This normalization procedure is only needed for the Xethru X2 SOC powered FlatEarth Ancho™ module as the normalization is not done internally and the Xethru® X4M03 module normalizes the data internally and gives out the normalized output through it's serial interface.

The Xethru®X4M03 module consists of a pair of directional patch antennas with integrated WiFi filter, Xethru X4 radar SOC, and an Atmel®SAMS70 microcontroller for controlling the X4 SOC and communicating with external devices connected through a USB interface. The X4 SOC also transmits a pulse and samples it through multiple parallel samplers similar to the X2 SOC. However, the X4 SOC is capable of generating bi-phase pulses (Pulses with 180 degree phase shifts) and an optional down-conversion step for filtering and decimating the sampled RF data to complex base-band data.

Both radar modules used in this research are compliant with FCC (Federal Communications Commission in USA) and ETSI (European Telecommunications Standards Institute) standards [14]. The FCC ID and Industry Canada (IC) certification numbers for the Xethru X4 radar module are: FCC ID: 2AD9QX4M02, IC Certification Number: 22782-X4M02. Table 2.1 presents some common differences and similarities between the two radar sensors and as seen on this table, both radar sensors have very similar features. However, even-though the research was started with the FlatEarth Ancho radar kit, the lower price and better performance of the Xethru X4M03 radar chip which was released in 2017 motivated the inclusion of the X4M03 radar in the final system and data collection experiments.

The data from both aforementioned radar modules are received as individual frames and each frame consists of an array of numbers with each number corresponding to the amplitude of the received signal at given time instances. These time instances correspond to distances as explained above and each frame is saved as a 2D Matrix with the first column corresponding to the timestamp at which the radar data was received. For the remainder of this thesis, the time

Table 2.1: Comparison of the two radar modules used for this research

Parameter	**Flatearth Ancho radar dev. kit**	**Xethru X4M03 radar**
Radar type	Ultra wide band	Ultra wide band radar
Bandwidth	6.0-8.5 GHz band	6.0-8.5 GHz band
Special features	Includes a Beaglebone Black single board computer (SBC)	Atmel ARM microcontroller for limited onboard processing
Connectivity	Can be connected to a computer through USB	Can be connected to a computer or a SBC
Range	Up-to 10m range	Up-to 10m range
Other comments	Sampling rate reduces with range	Sampling rate does not reduce with range
Price	US$ 3,700	US$ 399

	Fast time										
Slow time	Timestamp(ms)	Distance d1	d2	d3	d4	...	...	d(N-4)	d(N-3)	d(N-2)	d(N-1)	d(N)
	257.65	887140	887980	892800	893870	...	...	894400	889620	883990	884000	888820
	340.6	886690	887540	892410	893480	...	...	894120	889080	883550	883680	888740
	426.03	886860	887640	892390	893500	...	...	894330	889610	883820	883710	888520
	509.94	886640	887410	892170	893390	...	...	894370	889580	883650	883560	888480
	609.93	886470	887120	891810	892970	...	...	894400	889550	883680	883540	888290
	694.12	886360	887030	891890	893120	...	...	894260	888990	882980	883240	888450
	64917	885780	886450	891160	892340	...	...	892490	887970	883150	883670	888110
	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...
	65004	885640	886490	891140	892290	...	...	892370	887720	882970	883590	888230
	65088	885800	886550	891290	892390	...	...	892370	887820	882980	883580	888170
	65172	885770	886580	891230	892450	...	...	892470	887710	882870	883480	888270
	65257	885590	886370	891150	892370	...	...	892460	887780	882780	883420	888220

Figure 2.6: Data format used for collecting data from the radar module

	Fast time						
Slow time	Timestamp(ms)	Distance d1	d2	...	...	d(N-1)	d(N)
	254.12	7.0195e-06+2.1557e-06i	5.7537e-07+6.0299e-06i	...	...	-0.00021081+0.00019418i	-3.9731e-05+0.000357i
	269.9	6.1373e-08-2.1711e-05i	-2.6851e-07-2.3958e-05i	...	...	-0.00020656+0.00023399i	-2.6444e-05+0.00040573i
	276.64	3.7821e-06+5.3087e-06i	-1.6601e-05+5.3394e-06i	...	...	-0.00020371+0.00024688i	-3.7606e-05+0.0004245i
	281.08	-2.4434e-05-4.8791e-06i	-2.2593e-05-3.8435e-06i	...	...	-0.00021681+0.0002391i	-3.1691e-05+0.00041293i
	311.97	3.4522e-07-2.089e-05i	-9.812e-06-1.1362e-05i	...	...	-0.0001923+0.00022037i	-2.8845e-05+0.00039308i
	357.11	6.6052e-06+1.0679e-05i	-9.8196e-07+1.8542e-05i	...	...	-0.00020202+0.00022329i	-3.4054e-05+0.00038985i
	415.68	-5.5466e-06-2.57e-06i	-2.8431e-05-6.3751e-06i	...	...	-0.00019217+0.00021679i	-1.855e-05+0.00038089i
	...	...	...	...	...	...	...
	...	...	...	...	...	...	...
	65039	-2.4733e-05-2.9689e-06i	-2.7579e-05+4.9712e-06i	...	...	-0.00019696+0.00023495i	-1.5489e-05+0.00039882i
	65097	-3.4752e-06-1.4453e-05i	-1.723e-05-1.7576e-05i	...	...	-0.00016443+0.00021139i	-2.6467e-06+0.00037095i
	65159	-1.0142e-05-1.4162e-05i	-1.9816e-05-2.5777e-06i	...	...	-0.00020235+0.00023793i	-2.6275e-05+0.0003949i
	65222	-1.9946e-07+2.2585e-05i	-2.3467e-05+1.4215e-05i	...	...	-0.00016588+0.00021666i	-1.5059e-05+0.00038757i

Figure 2.7: Data format used for collecting data from the Xethru X4M03 module

instances within each frame are referred to as fast time and the times between each frame are referred to as slow time. An example of the data format corresponding to the X2 and X4 SOCs can be seen in Figures 2.6 and 2.7.

2.2 Noise and stationary clutter reduction

2.2.1 SVD based clutter reduction

The data generated using radar reflections include background clutter that makes detection of a person difficult without removing such clutter. Singular value decomposition (SVD) based clutter removal can be used for identifying and removing such clutter [15, 16, 17].

The radar data for a specific time window are organized as a $M \times N$ matrix where M is the number of recorded frames (slow time) and N is the number of points (range bins) in the fast time direction. The number of frames M is assumed to be greater than or equal to the number of range bins for a sufficiently large time window. Let X be the matrix of radar data organized as an $M \times N$ matrix, singular value decomposition of X is given by (2.16).

$$X = USV^T \tag{2.16}$$

Where U and V are $M \times M$ and $N \times N$ matrices respectively with $U^TU = V^TV = VV^T = I_N$, and S is a diagonal matrix consisting of the non-negative square roots of the eigenvalues of X^TX in decreasing order of value and they are called singular values.

$$S = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$$

Therefore, if $\text{rank}(A) = r, \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_N = 0$.The columns of U and V are called the left and right singular vectors. Equation 2.16 can be expanded as;

$$\begin{aligned} X &= \sigma_1 \begin{pmatrix} \vdots \\ u_1 \\ \vdots \end{pmatrix} \begin{pmatrix} \ldots v_1^T \ldots \end{pmatrix} + \sigma_2 \begin{pmatrix} \vdots \\ u_2 \\ \vdots \end{pmatrix} \begin{pmatrix} \ldots v_2^T \ldots \end{pmatrix} + \cdots + \sigma_r \begin{pmatrix} \vdots \\ u_r \\ \vdots \end{pmatrix} \begin{pmatrix} \ldots v_r^T \ldots \end{pmatrix} \\ &= \sum_{i=1}^{r} \sigma_i u_i v_i^T \\ &= M_1 + M_2 + M_3 + \cdots + M_r \end{aligned}$$

Where M_i are matrices of the same dimensions as X and are called as modes or ith eigenimage of X. X can be decomposed into three subspaces corresponding to clutter (M_c), target (M_t) and noise (M_n) [15],

$$X = M_c + M_t + M_n \tag{2.17}$$

Verma et al. in [15] states that M_c, M_t and M_n can be estimated by 2.18,2.19 and 2.20 respectively

$$M_c = M_1 = \sigma_1 \times u_1 \times v_1^T \tag{2.18}$$

$$M_t = M_2 = \sigma_2 \times u_2 \times v_2^T \tag{2.19}$$

$$M_n = \sum_{i=3}^{r} \sigma_i \times u_i \times v_i^T \tag{2.20}$$

Verma et al. in [15] mentions that the target signal is contained within the second Eigen image and therefore the target signal can be extracted by using the second Eigen image. However it was noticed that in most cases some of the target signal is also contained in the 3rd, 4th and other subsequent Eigen images and thus some of the signal is removed as noise when using only the second Eigen image. Therefore just removing the first Eigen image and using the remaining components provided better results overall than just using the second Eigen image. Figure 2.8 shows the raw signal from the radar before and after SVD based clutter removal. In the plots in Figure 2.8, the x and y axes are distance and amplitude respectively and each frame (a frame corresponds to a row in the radar data matrix) is drawn on top of each other. $i.e.$ time is drawn in the z axis.

The radar data shown on Figure 2.8 corresponds to a situation where the person is located at a distance approximately 1m away from the radar. As seen on this figure, the location where the person is in is clearly visible in the radar data after the SVD based clutter removal.

Verma et al. in [15] analyses the use of SVD based clutter removal and reports a PSNR value of 23.6dB for SVD based clutter removal. PSNR is a metric used to compare data before and after noise removal and [15] defines this metric by equations 2.21 and 2.22

$$PSNR(dB) = 10\log(1/MSE) \tag{2.21}$$

$$MSE = \frac{1}{M \times N}\sum_{i=1}^{N}\sum_{j=1}^{M}\{g(i,j) - f(i,j)\}^2 \tag{2.22}$$

Where, $g(i,j)$ is the original radar data matrix and $f(i,j)$ is the reconstructed radar data matrix after clutter reduction. And M and N are the dimensions of the radar data matrix as defined in the beginning of Section 2.2.1

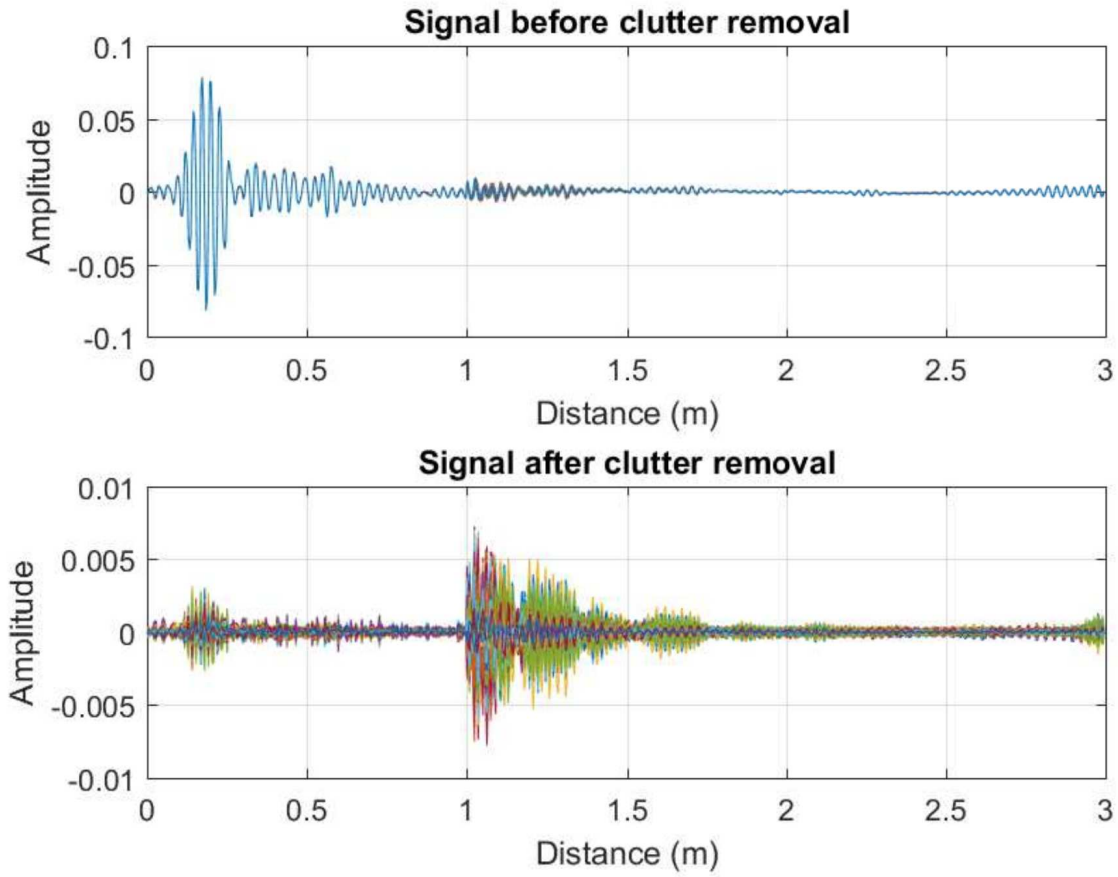


Figure 2.8: Radar data before and after SVD clutter removal

After the SVD based clutter removal the location of the subject can be simply identified by selecting the range bin corresponding to the maximum signal amplitude through time.

2.2.2 Identifying the presence of a breathing-like signal

[18, 19] proposes the idea of calculating the autocorrelation of the signal obtained from a candidate range bin (*i.e.* a single column from the radar data matrix defined in Section2.2.1) and

using the width of the autocorrelation main lobe centered around zero to identify the presence of breathing-like signals. The algorithm presented by Khan et. al. in [18] proposes that the autocorrelation main lobe width is approximately three times wider when the person is stationary and breathing compared to when the person is not-stationary. The publication shows two figures of the autocorrelation of the detected signal when the person is stationary versus moving and the figure corresponding to a stationary human shows the autocorrelation function of a perfect sinusoid. However, as shown in Section 3.1.2, the human breathing signal does not follow a perfectly sinusoidal waveform and a new method based on [18] is presented in Section 3.1.2.

H. Guo in [20] suggests using a neural network classifier trained on noise versus life signal in frequency domain to identify the presence of a breathing-like signal. The inputs to the NN classifier are 50 points of the frequency spectrum of the received signal and the network consists of a single hidden layer of 10 nodes. This network would perform well if the breathing signal is constrained to a fixed range of frequencies but its performance will degrade when the breathing signal is within a frequency range that the network was not trained on. The rationale behind this is that the network is a single layer NN with the inputs containing amplitudes of a fixed frequency spectrum and therefore the network is trained with higher values on a fixed range of values corresponding to human breathing and the normalized frequency spectrum of noise has higher values in different frequency ranges. However, if the breathing signal is of a higher than normal (or lower) frequency range that the network was not trained on, the performance of the NN will deteriorate as the network has not seen respiration signals corresponding to these frequency ranges. A convolutional neural network trained to identify frequency domain features irrespective of their position would perform better. The author does not mention this issue in their publication and the work mainly focuses on the error rate of classification corresponding to different signal to noise ratios.

2.3 Breathing rate estimation

After identifying the range bin where a human target is located, the next task is to estimate the frequency of the breathing signal present in the selected bin.

2.3.1 FFT based breathing rate estimation

Analyzing the Fourier Transform of a signal is one of the most popular methods of estimating the repetition rate of a given signal. [21] uses many FFT[7] based methods for phase and frequency estimation and also presents zero-crossing phase and frequency estimation methods.

If the signal captured from the UWB radar contained a pure sinusoidal breathing signal, estimating the breathing rate would be as simple as estimating the frequency of this sinusoidal signal by finding the frequency with the maximum amplitude on the Fourier transform of the signal. But as presented in [17] the human breathing signal captured using a PN-UWB radar contains several harmonics of the fundamental breathing signal. [17] also mentions that the second harmonic of the breathing signal can be higher than the first harmonic of the signal and thus a simple method of taking the frequency value with the highest magnitude would lead to obtaining a breathing rate that is twice the breathing rate of the subject. Since both the abdomen and the chest move when a human is breathing, the signal captured contains components from both of these movements. As presented in [17], the signal captured from a UWB radar can be modeled using equation 2.23 where $m_c(t)$ and $m_a(t)$ are the movements of the chest and abdomen respectively and, d_0 is the distance to the target from the UWB radar. Further, c_i are the harmonic components of chest displacement and a_j are harmonic components of the abdomen displacement and f_b is the breathing frequency.P and Q represent the number of significant components of the chest and abdomen displacement respectively.

$$\begin{aligned} d(t) &= d_0 + m_c(t) + m_a(t) \\ &= d_0 + \sum_{i=1}^{P} c_i sin(2\pi i f_b t) + \sum_{j=1}^{Q} a_j sin(2\pi j f_b t) \end{aligned} \tag{2.23}$$

As seen from equation 2.23, since the signal contains harmonic components of the fundamental breathing frequency, summing up multiple harmonic components of the breathing signal after applying some *"operation"* to the frequency components and getting the frequency component with the maximum magnitude value would result in obtaining the fundamental breathing frequency. This common step behind most FFT based breathing rate estimation algorithms is presented in Algorithm 2.1

The *operation* applied in the function **FN** in Algorithm 2.1 is as follows:

[7] Fast Fourier Transform

Algorithm 2.1 Common operation behind most FFT based breathing rate estimators.

Input:

SIGNAL: Time-domain signal corresponding to the subject location

1: DFT = FFT(SIGNAL) ▷ DFT stands for the discrete Fourier transform spectrum
2: **for** $i \leftarrow 1$ to $\frac{1}{K} \times$number of FFT points **do**
3: result$(i) = \sum_{j=1}^{K}$FN(DFT$(j \times i)$)
4: **end for**
5: **function** FN(x)
6: result = apply some *operation* to x
7: **return** result
8: **end function**

- Rectangular moving window algorithm: select a window of four points around x (*i.e.*select $DFT(i \times j + l)$ where $l = -2, -1, 0, 1, 2$)and sum up the values in the window.
- Sum squared algorithm: $result \leftarrow x^2$, *i.e.* $result = DFT(i \times j)^2$
- Shannon energy based algorithm: $result \leftarrow -x^2 \times \log(x)$ where $x = DFT(i \times j)$
- Log entropy based algorithm: $result \leftarrow \log(x)^2$ where $x = DFT(i \times j)$

A FFT based Ellipse fitting algorithm introduced in [22, 23] and used by [24] has also been tested in Chapter 4 and even though it gives satisfactory results for time windows of around 60 seconds or more (This corresponds to the total time of the signal and not the shift window of the ellipse fitting algorithm as described in [24]), when the time window gets narrower, the results degrade and tend to pick up sub-harmonics. Chang Li in [24] uses the Welch periodogram introduced by P. Welch in [25] to identify the breathing frequency by observing the frequency corresponding to the maximum peak in the Welch Periodogram. The rationale behind this is because the Welch periodogram performs better than purely Fourier transform based methods in instances when the signal is non-stationary because the Welch periodogram is calculated by time averaging over short, modified periodograms [25]. This method has also been explored in this thesis and the results are available in Section 4.3. A notch filter based algorithm where a notch filter is applied to remove harmonic components of a possible breathing rate as described in [17] was also tried out and the results can be seen in Section 4.3.

Even-though these algorithms perform relatively well when the breathing signal follows the model defined in equation 2.23, when sub-harmonics of the breathing rate are present in the breathing signal, the frequency summation algorithms similar to algorithms defined by Algorithm

2.1 fail as they tend to amplify sub-harmonics instead of the component corresponding to the actual breathing rate.

2.3.2 Zero-crossing method for breathing rate estimation

The zero-crossing algorithm presented in [17] is different from the pure frequency domain algorithms because it analyzes the signal in the time domain and determines the cut-off frequencies for a band-pass filter to apply on the signal as shown in algorithm 2.2. The signal after applying the band-pass filter is analyzed in the frequency domain to obtain the breathing rate in this algorithm. A simplified description of the zero crossing algorithm is shown in algorithm 2.2

Algorithm 2.2 Simplified Zero-Crossing algorithm

Input:
Signal $y(t)$ of range bin corresponding to the person location.
Threshold multiplier $\beta(typically = 1)$

while (mean time between crossing points $\neq$ time corresponding to frequency with maximum magnitude after adaptive bandpass filter $\pm$ constant) **do**
  Positive threshold $\alpha_p \leftarrow \beta \times StdDeviation(y(t))$
  Negative threshold $\alpha_n \leftarrow -\beta \times StdDeviation(y(t))$
  $t_{upper} \leftarrow$ indices where $y(t) > \alpha_p$
  $t_{lower} \leftarrow$ indices where $y(t) < \alpha_n$
  Remove adjacent time values from t_{upper} and t_{lower}
  $Pt_{ZC} \leftarrow$Combine t_{lower} and t_{upper} and sort according to ascending t and find zero crossing points between points in t_{upper} and t_{lower}
  $\mathcal{S}_{PTP} \leftarrow$ interval between points in Pt_{ZC}
  $\mathcal{S}_{PTP} \leftarrow$ Choose only intervals that are ≤ 11 and ≥ 1.5 seconds
    ▷ The human breathing pulse can not have a period less than 1.5 seconds or more than 11 seconds [17]
  $T_B \leftarrow$ mean of $\mathcal{S}_{PTP}$
  $f_B = \frac{1}{T_B}$
  $\gamma = 1.2$ ▷ Section 5.3.2.1 of [17] states that a γ value of 1.2 gives the lowest mean absolute error for breathing rate estimation
  $f_u = \frac{1}{T_B - \gamma\sigma_B}$
  $f_l = \frac{1}{T_B + \gamma\sigma_B}$
  $y(t) \leftarrow$Apply a 6$^{\text{th}}$ order IIR Butterworth bandpass filter with upper cut-off f_u and lower cut-off f_l to $y(t)$
  $f_{bpf} \leftarrow$ frequency with maximum magnitude after applying bandpass filter.
  if $|f_b - f_{bpf}| < 0.05$ **then**
    return breathing frequency f_{bpf}
  else
    continue ▷ Note that $y(t)$ is different in each iteration due to the changing cut-off frequencies of the bandpass filter.
  end if
end while

This algorithm also does not provide satisfactory results and in addition, the time spent within the outer while loop as described in Algorithm 2.2 is heavily dependant on the quality of the input signal and thus the run-time of the algorithm can not be reliably estimated for a generic input and tends to vary heavily (between 0.4 seconds to a few seconds or more on Matlab® on an Intel® Core i7™ 6700HQ processor based system with a 16GB of RAM) and thus was not used for real-time estimation of breathing.

2.3.3 Empirical Mode Decomposition based breathing rate estimation

Empirical Mode Decomposition is a method for analyzing nonlinear and non-stationary data by decomposing the data into a finite number of intrinsic mode functions that can be effectively analyzed through Hilbert transforms as the concept of instantaneous frequency makes sense on these intrinsic mode functions [26].

An intrinsic mode function (IMF) is a function that satisfies two conditions:

1. The number of extrema and the number of zero crossings must either be equal or differ at most by one

2. At any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero

The notion of instantaneous frequency presents a controversy as the frequency is usually defined for the sine or cosine function spanning the whole data length with constant amplitude. This means that the instantaneous frequencies also have to relate to either a sine or a cosine function and thus at least one full oscillation is needed to define a local frequency value. For an arbitrary time series $X(t)$ the Hilbert Transform $Y(t)$ is defined as;

$$Y(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{X(t')}{t - t'} dt' \tag{2.24}$$

where P indicates the Cauchy principal value. With this definition $X(t)$ and $Y(t)$ form the complex conjugate pair, so we can have an analytic signal $Z(t)$ as

$$Z(t) = X(t) + iY(t) = a(t)\mathrm{e}^{i\theta(t)} \tag{2.25}$$

where,

$$a(t) = [X^2(t) + Y^2(t)]^{1/2}, \quad \theta(t) = arctan\left(\frac{Y(t)}{X(t)}\right) \tag{2.26}$$

The instantaneous frequency of a signal can then be defined as shown in equation 2.27 where $\theta(t)$ is the instantaneous phase of the signal.

$$\omega = \frac{d\theta(t)}{dt} \tag{2.27}$$

As the instantaneous frequency given in equation 2.27 is a single value function of time, at any given time, there could be only one frequency value. This introduces the concept of a 'monocomponent' signal and according to [26], the intrinsic mode functions obtained after empirical mode decomposition involves only one mode of oscillation. Therefore an IMF is not restricted to a narrow band signal but still a signal that can be used to evaluate instantaneous frequencies. Hence, EMD can be used for decomposing the radar reflected signal from a human and identify its individual components such as heart beat, respiration, noise etc. In addition this also allows the calculation of instantaneous frequencies of such signals.

The EMD algorithm works by extracting intrinsic mode functions step by step in an empirical manner. The process of extracting an IMF involves a procedure called sifting. The sifting process is as follows:

1. Identify all the local extrema in the data
2. Connect all local maxima by a cubic spline as the upper envelope
3. Repeat the same procedure for local minima to produce the lower envelope

If the mean of the upper and lower envelopes is m_1, the difference between the data and m_1 is the first component h_1

$$X(t) - m_1 = h_1$$

In the next step the same procedure is repeated after treating h_1 as data

$$h_1 - m_{11} = h11$$

After repeated sifting up to k times, h_1 becomes an IMF,

$$h_{1(k-1)} - m_{1k} = h_{1k}$$

Then, h_{1k} is designated as the first IMF component of the data, i.e. $c_1 = h_{1k}$

The stopping criteria of the sifting process as proposed in [26] is to stop when the standard deviation is smaller than a pre-given threshold. The definition of standard deviation is given in equation 2.28

$$SD_k = \sum_{t=0}^{T} \frac{|h_{k-1}(t) - h_k(t)|^2}{h_(k-1)^2(t)} \tag{2.28}$$

Jian et al. in [27], have used EMD to extract heartbeat and breathing signals from UWB radar returns. Their work focuses on reconstructing the heartbeat and respiration signal by combining multiple intrinsic mode functions that can be candidates for a respiration or heart beat signal by observing their frequencies. This work however does not propose a method to automatically select the candidate IMFs and does not present a reference to confirm that the extracted signal is in fact the correct respiration/heart signal. [27] also suggests using a band-pass filter after obtaining the combined signal from several IMFs which is used to remove frequencies out of the heartbeat/respiration range.

Sun and Li in [28] purposes using EMD to detect life in coal mines. After decomposing the radar reflected signal, their work focuses on Fourier analysis of each IMF to detect the dominant frequencies of the IMFs. The presence of life is characterized by the presence of frequency components that are in frequency ranges typically associated with breathing or heartbeat signals. Their work also does not focus on selecting the correct IMF automatically and selects the IMFs manually. A major drawback of their method is the inability to confirm if the selected IMF is actually a human breathing signal or a noise signal falling within the same frequency range. This would inhibit their ability to give accurate predictions of the presence of life in real-world situations.

Narayanan in [29] presents the use of EMD for breathing signal detection. However, their work selects the IMF corresponding to breathing manually and then calculates the breathing rate by counting the number of peaks of the signal above a manual threshold. They mention the use of Hilbert Spectral analysis to calculate instantaneous frequencies but does not use these values to estimate the breathing rate. Since the thresholds are also selected manually in addition to the

IMF corresponding to breathing, this algorithm relies heavily on human intervention to obtain the breathing signal and breathing rate.

Labate et al. in [30] compares EMD vs DWT(Discrete Wavelet Transform) based methods to extract the breathing signal they apply the discrete wavelet transform with different mother wavelets such as Daubechies (3 and 6) and Symlets (3 and 6), and claims that the correlation between the reference signal and the signal after wavelet transform is highest (93.01%) among all the signals after wavelet transform when the 6^{th} order Symlets wavelet is used. They also claim that EMD method gives better results than wavelet transform with a correlation of 93.07% with the reference signal. However, they do not introduce a method to select the correct IMF(s) corresponding to the breathing signal when using the EMD method and do not address the fact that the discrete wavelet transform has the ability to automatically give the corresponding signal in its approximation coefficients. Hence, the wavelet transform here has the advantage of automatically giving the corresponding signal compared to the EMD method as with EMD method, they choose the corresponding IMF manually. However, an issue with the Wavelet transform is that the choice of wavelet used affects the output signal and some priori knowledge about the signal is assumed before performing the wavelet transform. The priori assumption here usually is that the breathing signal is periodic and resembles a sinusoidal signal up-to some extent. However, as seen in Figure 2.9 breathing signals are usually not periodic and not sinusoidal and the EMD method usually gives a better approximation of the breathing signal than the DWT based method. The work in this thesis uses the fact that wavelet transform gives a sufficient approximation of the breathing signal in order to select the correct IMF(s) so that a better approximation of the breathing signal can be found automatically.

All the algorithms/methods [27, 28, 29, 30] presented above in this section lack the ability to select the correct[8] IMF from the radar return signal. In this thesis, a novel method is presented in 3.2.2 that can select the correct IMF automatically.

Xinyang Zhang in [31] suggests an algorithm to automatically select the IMF corresponding to breathing after empirical mode decomposition of the signal. The IMF is selected by evaluating the Minkowski distance which is defined by $d_{mink} = \left(\sum_{i=1}^{n} |sig_i - imf_i|^2\right)^{1/2}$ where sig_i and imf_i are the i^{th} respective samples of the observed radar signal and the extracted IMF

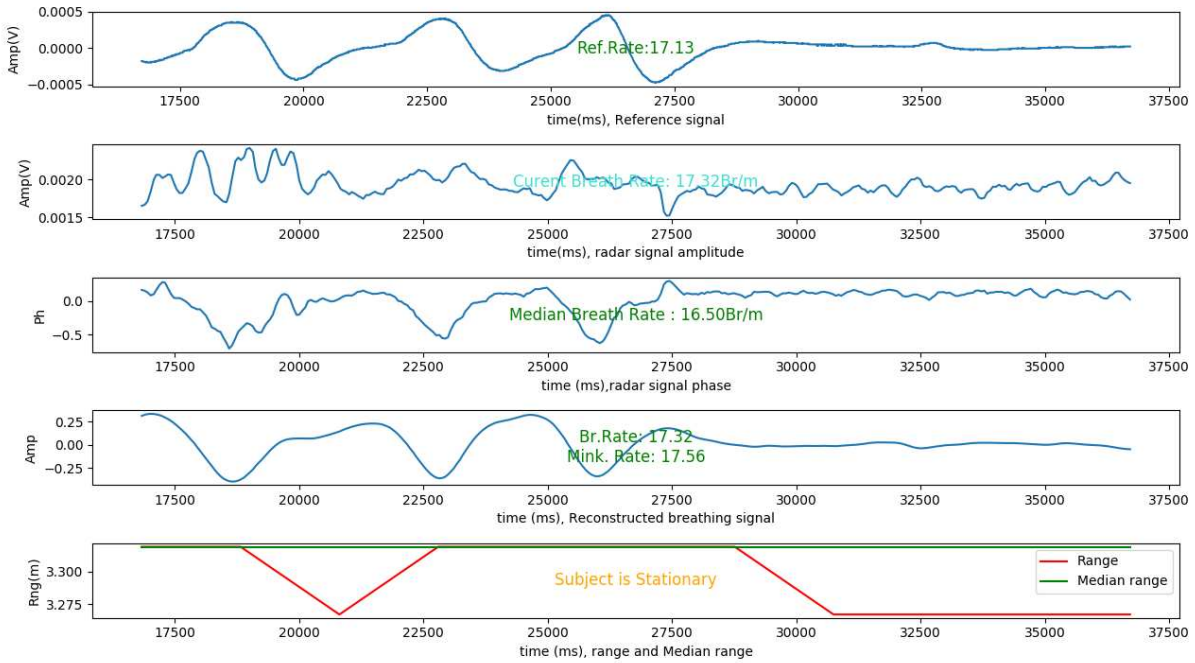
[8]Throughout this thesis the term correct IMF is used for IMF(s) with dominant oscillation rate sufficiently close to the reference breathing rate obtained using a reference source

respectively and n is the number of samples in time domain. The IMF which demonstrates the minimum Minkowski distance is selected as the breathing signal and a simple Fourier analysis of the signal gives the breathing rate in this algorithm. Even though this algorithm is able to select the correct IMF in most cases, this algorithm fails in cases such as that is shown in Figures 2.10 and 2.11. Figures 2.10 and 2.11 each shows five plots. The first plot shows the breathing signal captured using a respiration belt which is used as the reference source and the second and third plot shows the amplitude and phase of the signal captured using a Xethru® X4 radar sensor module (These amplitude and phase signals corresponds to the signal derived in Equation 2.12). The fourth plot shows the reconstructed breathing signal using the novel algorithm described in Section 3.2.2 and the text overlay on the fourth plot shows the breathing rate calculated using the novel algorithm as "Br.Rate" and the breathing rate calculated using the Minkowski distance based method in [31] is shown as "Mink.Rate". As it can be seen here, the Minkowski distance based method has selected a sub-harmonic of the breathing rate and a higher frequency component (Possibly a harmonic component of the breathing signal) in the two cases respectively.

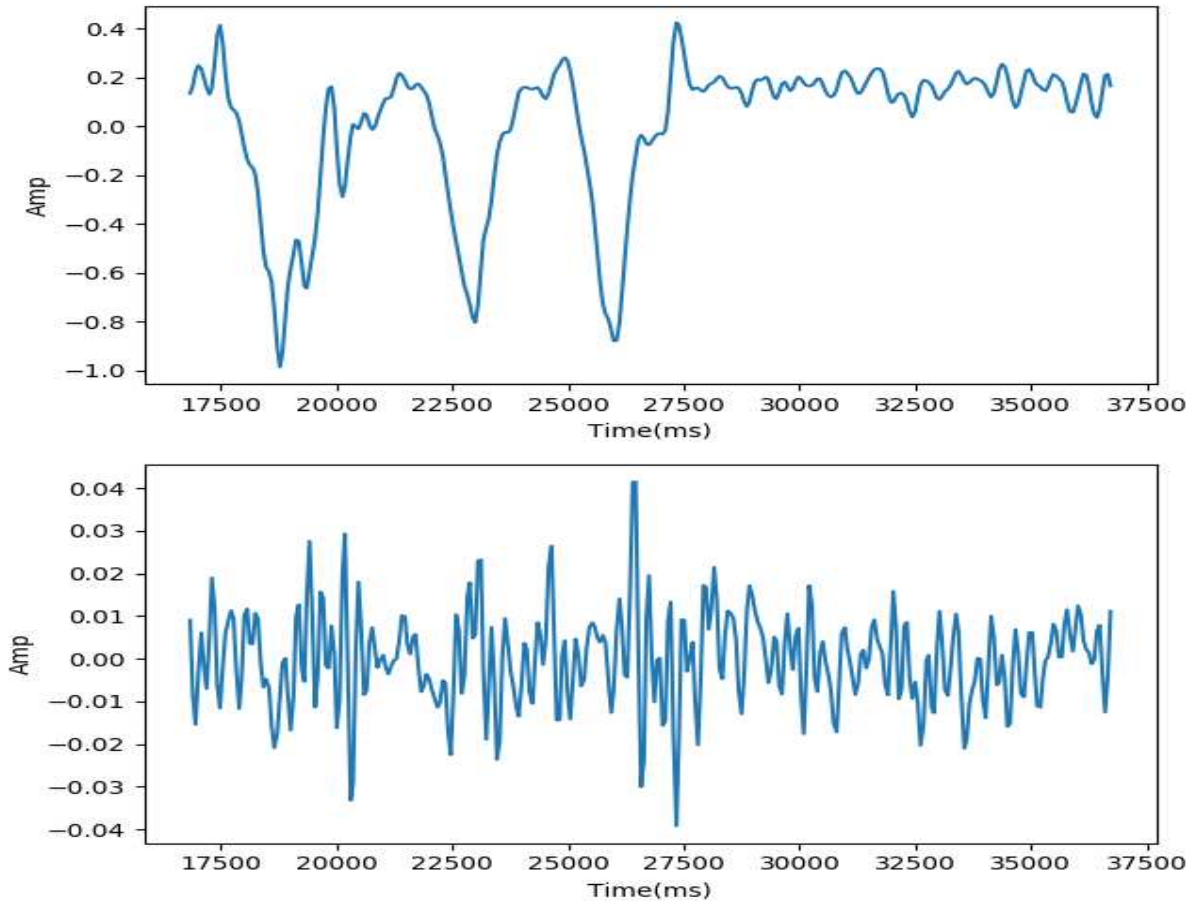
Figure 2.12(a) shows the breathing signal captured using the radar along with the breathing signal from a respiration belt sensor. Figure 2.12(b) is organized into three columns of sub-figures where the first column is the time domain representation of each IMF and the second column shows the frequency domain representation of each IMF along with a text overlay that shows the breathing rate obtained from each IMF and the Minkowski distance calculated according to [31]. The third column shows the instantaneous frequency of each IMF calculated using Hilbert Spectral Analysis. As seen on Figure 2.12(b), the Minkowski distance of the IMF with 2.26 Br/m as the breathing rate is the lowest with a value of 12.01. The reference rate obtained in this case is 17.3 breaths per minute and it can be seen on Figure2.12(b) that the empirical mode decomposition has decomposed the breathing signal into two modes with fundamental oscillation rates of 19.3 and 16.3 breaths/minute. This is another shortcoming of Xinyang Zhangs method [31] as their method does not take into account the fact that the breathing signal could be decomposed into two IMFs by the EMD algorithm. Thus it is clear that the Minkowski distance based method introduced in [31] does not correctly select the IMF corresponding to the breathing signal in this case as well.

Further results obtained using the Minkowski distance based method and other methods can be found in Section 4.3 and an improved method based on the Complete Ensemble Empirical Mode Decomposition (CEEMD) with a novel scoring algorithm to select the correct IMF has been introduced in this thesis in Section 3.2.2 and it's results are compared with other existing algorithms in Chapter 4

[9]Further details about this figure can be found in Section 2.3.3



(a) From top to bottom: (1) Reference signal captured using a respiration belt, (2) & (3) amplitude and phase signals captured using the radar, (4) reconstructed breathing signal using the EMD based method in Section 3.2.2, (5) distance to the person from the radar through time



(b) Approximation (top) and detail(bottom) signal extracted using DWT

Figure 2.9: Comparison between the signal reconstructed using EMD and DWT based methods

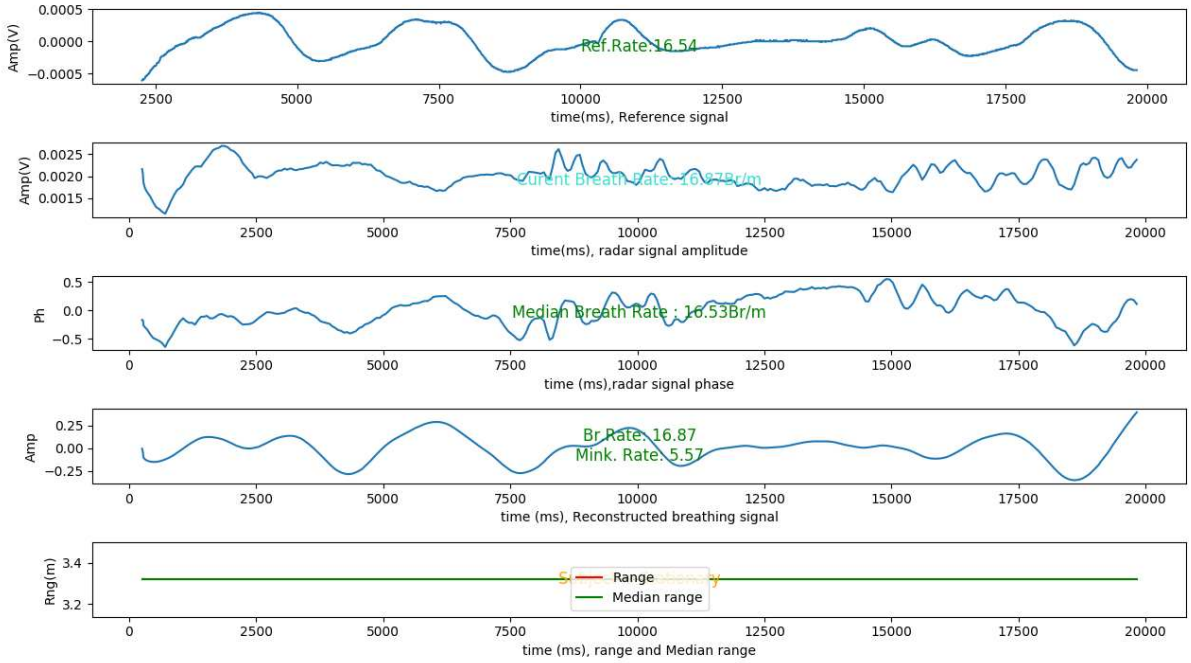


Figure 2.10: A case when the Minkowski distance based method selects a sub harmonic as the breathing signal(An explanation about this figure can be found in the text)

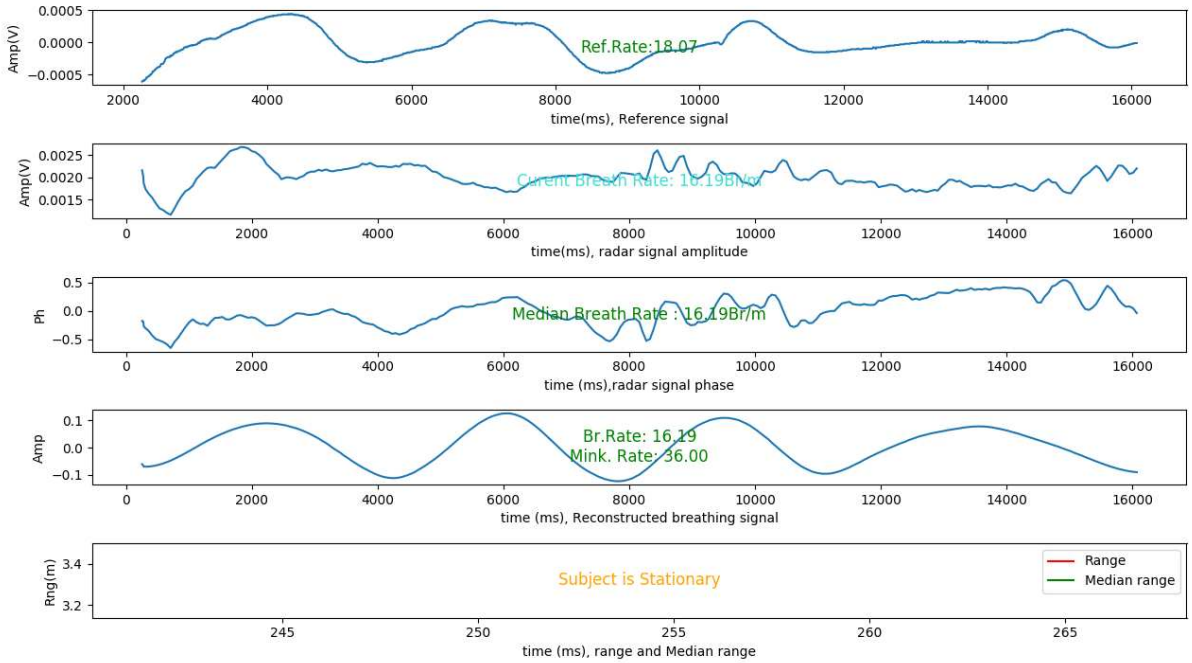
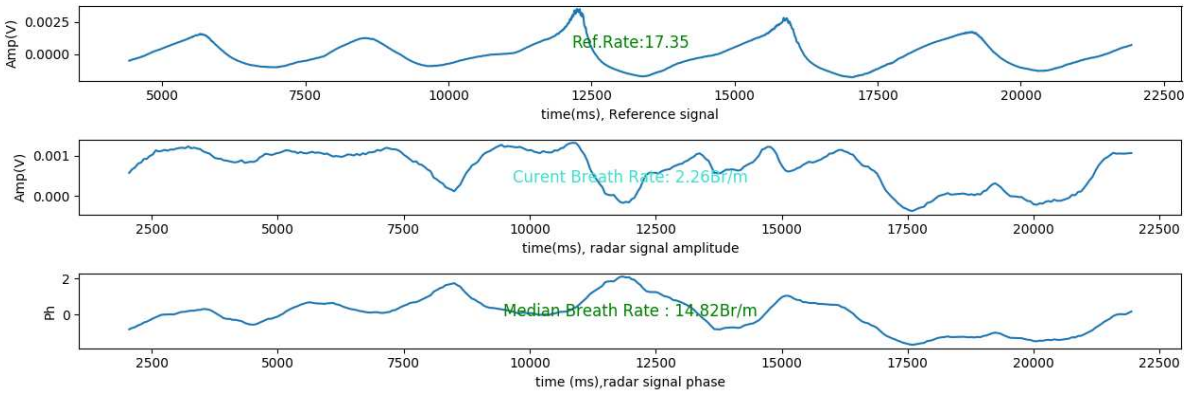
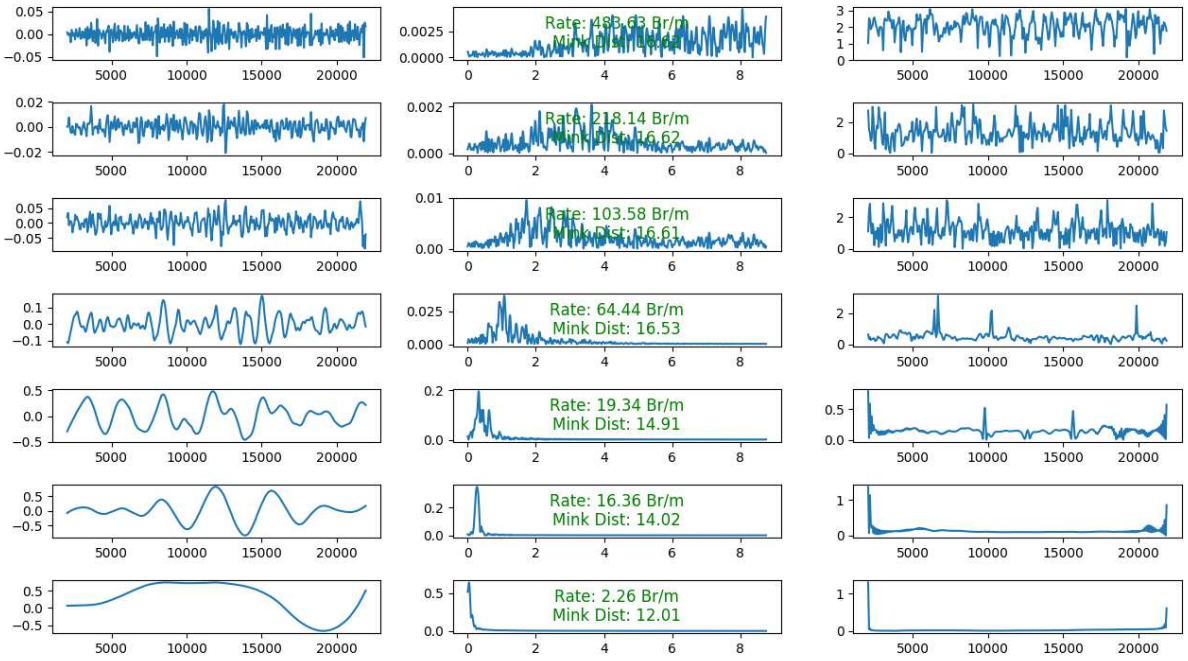


Figure 2.11: A case when the Minkowski distance based method selects a IMF corresponding to a higher frequency signal(*i.e.* Mink Rate in third plot is 36.0)(An explanation about this figure can be found in the text)



(a) Breathing signal captured by the radar and the reference signal from respiration belt



(b) Decomposed IMFs of breathing signal

Figure 2.12: Breath signal and it's intrinsic mode functions along with their Minkowski distances with the minimum Minkowski distance corresponding to an IMF that is not the breathing signal[9]

2.3.4 Other methods to estimate breathing

Ravichandran et al. in [32] uses four algorithms, (1)Zero-crossing detection, (2)Fourier transform maximum selection, (3) Linear predictive coding (LPC) and (4)Least-squares harmonic analysis(LSH) to estimate the breathing signal using two Universal software radios with one transmitting a continuous signal at 2.4GHz and the other receiving the reflected signal from the environment. According to their research breathing alters the magnitude of the reflected signal introducing an amplitude modulation on the transmitted wireless signal which can be expressed as,

$$u(t) = A_c m(t) \cos(2\pi f_c t) \tag{2.29}$$

where A_c and f_c are the amplitude and frequency of the 2.4GHz carrier signal and $m(t)$ is the breathing signal acting as a modulator on the carrier.

Zero-crossing detection method directly estimates the frequency of a periodic signal by measuring the number of negative-to-positive(or vice versa) transitions of a time waveform in a given time window. A very similar method is to detect the number of local maxima/minima of a time waveform in a given time window. In a very controlled case, these methods should work very well. However, in the presence of noise these methods which only rely on observing the time-domain properties of the signal will fail as they will not be able to effectively differentiate between noise and meaningful signals.

Fourier transform maximum selection is simply selecting the frequency corresponding to the maximum peak in the frequency domain. This has been used widely [18, 19, 33] to estimate breathing rate and gives promising results in controlled environments [32].

Ravichandran et al. in [32] combines the results from multiple algorithms which estimate the breathing rate once every 900 milliseconds. Their combination algorithm first determines the mean of the two closest frequency estimates between all the four algorithms, and once 5 past estimates are collected, they select the value that is closest to the median of the last 5 algorithms. The assumption they make is that since 5 estimates will take only 4.5 seconds as each estimate is calculated once every 900 milliseconds, a persons' breathing rate would not change significantly during a 4.5 second period. However, due to the nature of their algorithm which assumes at least two algorithms gives values close to the actual breathing rate, when only 1 algorithm works properly and the 3 remaining algorithms fail, their combination algorithm will still fail as it will

select two algorithms that give wrong results to calculate the mean breathing rates instead of the one algorithm that works well. Other than this shortcoming, their algorithm should work well when at least two algorithms give proper breathing rates. A better algorithm for combining multiple algorithms is presented in Section 3.2.4 which was developed by extending the work by Ravichandran et al.

Chapter 3

Methodology

Even though there are many algorithms for estimating the breathing rate using radar sensors, the lack of proper analysis of each algorithm against one another with a large set of data meant that most of these published algorithms work in specific situations that the author has disclosed but there was no framework to test many algorithms together in many situations. In order to properly evaluate each algorithm, a data collection software with the ability to collect data using a radar and a reference source (Respiration belt sensor or a reference signal from a linear actuator) along with video recordings of the test environment was developed using Matlab® and C++ along with the ability to process radar data and estimate breathing in real-time. The collected data is then analyzed using an algorithm analysis and report generation framework which loads up the radar data along with the reference data and creates reports after evaluating the performance of the algorithms. In order to speed up the real time estimation of breathing rate, the real-time component of the aforementioned Matlab® algorithm was implemented using Python and subsequently improved with more features and the ability to increase the frame rate of the radar data acquisition. This chapter describes the method of operation of these software components along with the algorithms used/developed for breathing estimation and ranging.

The four main applications developed are as follows:

1. Matlab® and C++ based synchronized data collector capable of collecting radar data from a FlatEarth® Ancho module or a Xethru® X4 radar module and a reference signal from a BioRadio™ 150 breathing belt sensor by GLneurotech® with relative timestamps for each frame from all sensors along with synchronized video data.

2. Matlab application to run multiple test-cases recorded by the data recording application in item 1 and generate LaTeXreports along with Microsoft® Excel sheets containing information about the performance of each algorithm.

3. A Matlab® real-time application with the ability to process radar data captured using a FlatEarth® Ancho module and produce breathing rate estimations along with breathing signal corresponding to the chest movement of the person during breathing.

4. A Python application with the ability to capture and process radar data using a FlatEarth® Ancho module or a Xethru® X4 module in real-time and run multiple breathing rate and breath signal detection algorithms using Matlab® or/and Python simultaneously and generate results.

Figure 3.1 shows the four applications categorized according to their ability to collect radar data and process them while collecting or offline after collection.

In addition to being able to run multiple existing algorithms to estimate breathing rate, a novel algorithm with better results than the existing algorithms was developed based on the Complete Ensemble Empirical Mode Decomposition (CEEMDAN) as described in Section 3.2.2 and a novel scoring algorithm was introduced to select the best performing algorithm dynamically as described in Section3.2.4. The Python based real-time application described in Section 3.6 includes all these features and is able to estimate the breathing rate of the subject in real-time.

In addition to the details about the applications developed during this research, this chapter also describes the processes involved in estimating breathing rate from a radar signal. The following steps are used for this task and the sections describing them are organized in the order in which the radar data are processed.

1. Variance based range estimation as described in Section 3.1.1

2. Identifying the presence of a breathing-like signal (Section 3.1.2)

3. Breathing rate estimation (Section 3.2). This step is done using the following steps with the different algorithms running parallely to estimate breathing rate in real-time

    - FFT based breathing rate estimation (Section 3.2.1)

Figure 3.1: Applications developed for data collection and processing

- EMD based breathing rate estimation (Section 3.2.2)
- Non-linear mode decomposition (NMD) based breathing rate estimation (Section 3.2.3)
- Automatically selecting the best performing algorithm dynamically to report breathing rate (Section 3.2.4)
- In addition to the three algorithms mentioned above, the Python and Matlab® Engine based system described in Section 3.6 is capable of computing the breathing rate using many other algorithms described in Chapter 2. However, because of the low accuracy of these algorithms, only the algorithms mentioned above are enabled in the final system. Results with all algorithms enabled are shown in Section 4.3 for reference.

3.1 Range estimation of a person and identifying the presence of a breathing-like signal

The Xethru X4™ and FlatEarth Ancho™[1] radar modules produces data frames as described in Section 2.1.1 when each frame is received, a time-stamp is recorded and stored along with the radar data frame so that the time at which the data was received and the actual frame-rate of the radar data can be calculated in real-time. The data format after this addition is described in Section 2.1.1.

3.1.1 Variance based range estimation

The assumption of $M \geq N$ in Section 2.2.1 in order to remove clutter from the radar signal introduces a limitation where SVD based clutter removal can not be used in near real-time radar data as M corresponds to the number of frames in time that the radar data needs to be collected for before removing clutter using SVD. As a person can be moving around during the collection of radar data in a real-world scenario when the radar is used, the SVD based clutter removal will fail to remove the undesired signals and estimation of a range bin will also fail as the person will be in multiple range bins throughout the time of the experiment.

[1] FlatEarth Ancho is a radar module developed by FlatEarth® using the Xethru X2™ radar SOC

For short time periods, looking at the variance of the signal in different range bins through time was used in order to properly estimate the location of a person and this method provides a fast and less resource intensive way to track a persons' location in a real-time application. [34] uses variance based methods to identify the location of a person using radar return signals. However, this work does not cover the instances when the person is not stationary and would work only when identifying the person's location when the person has been stationary for the entire period of time when the radar data was collected.

The methodology behind the variance based range estimation can be found in the flowchart shown in Figure 3.6. The system continuously collects data and normalizes them and after a set of new data corresponding to a time period of **T1** seconds is collected (Values for **T1** between 1 to 3 seconds were used in this work), the variance along the slow time axes of each range bin is calculated separately. A typical set of values for variance can be seen in Figure 3.2. Figure 3.2 shows the variance of the radar signal through slow time when the person is at a range approximately 1m from the radar. As it can be seen from this figure, the person can be very prominently seen through the variance value of the radar signal. The highest value of variance is selected through each set of slow time samples corresponding to different range bins. This range is then saved in a double ended queue (Q1) and the data on Q1 are analyzed to detect movements.

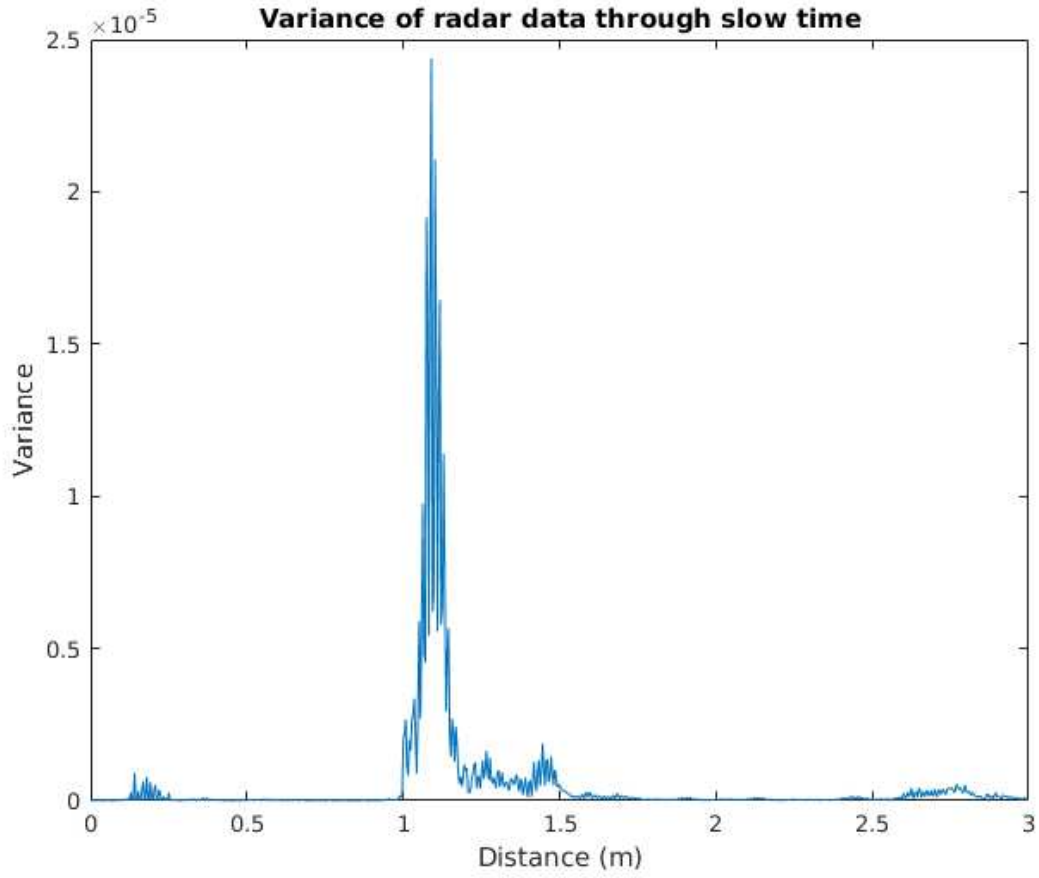


Figure 3.2: Variance along slow time axis of all range bins when a single person is detected

3.1.2 Identifying the presence of a breathing-like signal

Detecting the presence of a breathing-like signal is critical in order to estimate the breathing rate. As movements of the person being monitored other than chest movements due to breathing are captured by the radar and if these movements also fall within the same range-bin as the persons chest, the chest movements captured by the radar are modulated on top of the other movements in the same range-bin. In order to detect if such undesirable movements are present in the captured signal, [18] suggests looking at the autocorrelation of the signal in the selected range-bin. [18] suggests measuring the width of the main lobe between the points that it crosses 0. However, the auto-correlation function of a signal reflected from a stationary human does not look like what is presented by Khan et. al. in realistic situations. The figure presented in [18] is the autocorrelation function of a perfect sinusoidal signal (Figure 3.3). However, this is not the case in realistic situations. The auto-correlation function of a signal reflected from a stationary

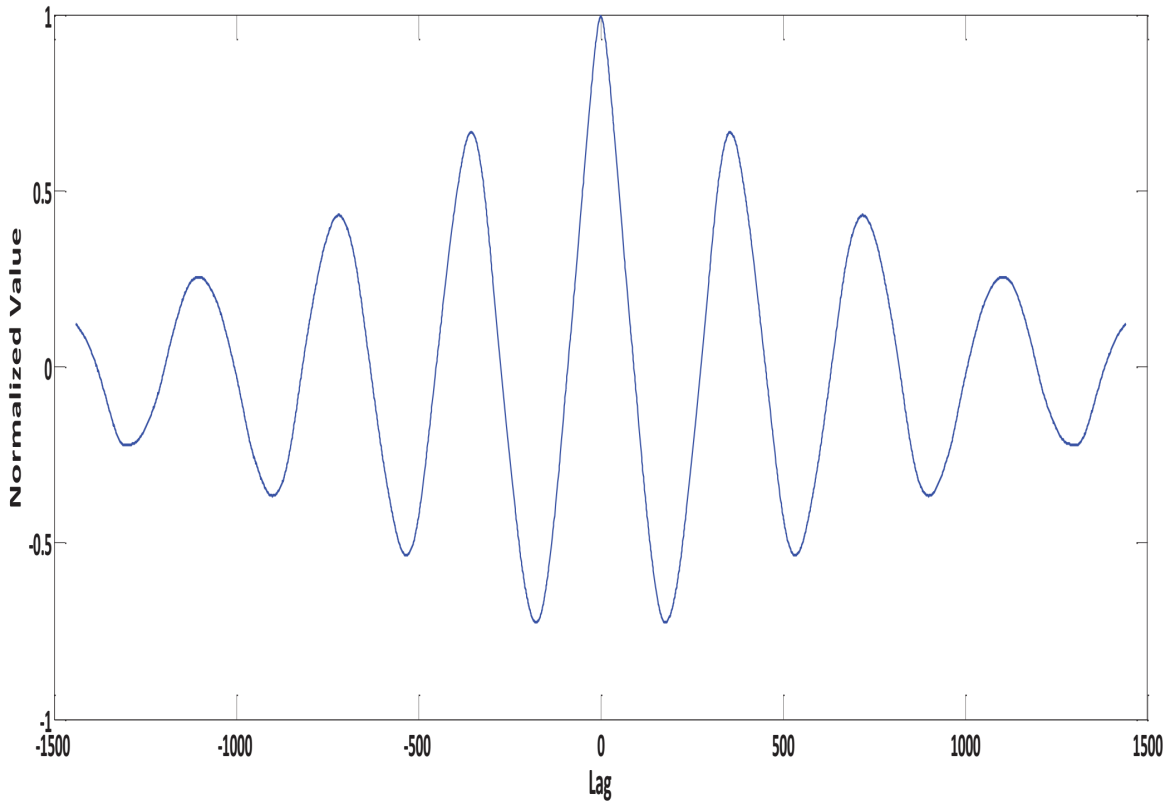


Figure 3.3: Figure presented by Khan et al. as the autocorrelation function of signal reflected from stationary human

human usually looks like what is shown on Figure 3.4. Figure 3.5 shows the auto-correlation function when a person is moving slightly. If only the width of the main lobe at points where it crosses zero is considered, both cases in Figures 3.4 and 3.5 would be considered as a person being stationary. Therefore in this thesis, the main lobe was identified by looking at the local minima's around 0 of the auto-correlation function.

Another issue with the method presented by Khan et. al. in [18] is that the presence of a breathing signal is only identified by looking at instantaneous signals. However, in a system that works continuously, it is possible to keep track of the location of the person being monitored and identify if they are moving. Therefore, two new conditions are introduced on top of the auto-correlation based method to identify movements of the person by looking at the history of their position as follows:

1. Identify range of a person once every **T1** seconds and store these values in a double ended queue (Q1) as described in Section 3.1.1

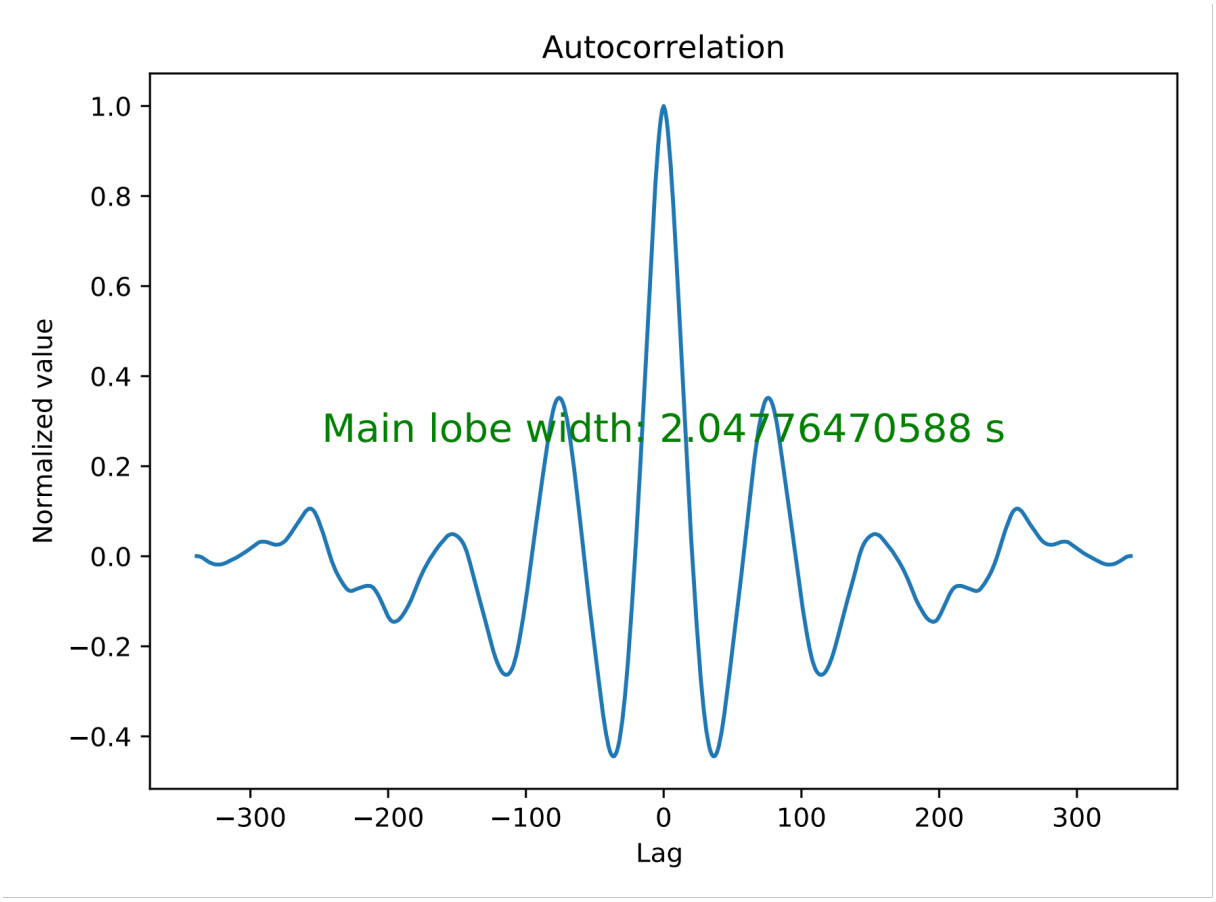


Figure 3.4: Autocorrelation function of a usual signal reflected from a stationary human

2. Calculate the difference between the two latest values in Q1

3. If the values are different than a preset threshold **TH1**, check if such differences were present for the last three consecutive estimates in Q1. If this is the case, classify this as the person moving around fast. If the difference between the latest values is less than **TH1**, move to next step.

4. Calculate the medians M1 and M2 of X1 and X2 latest values of Q1, where X2»X1. (X1=3 and X2=20 in the results presented in this work)

5. If difference between M1 and M2 is greater than threshold **TH2**, classify as a movement. If not, go to next step.

6. Check if a breathing like signal is is present by examining the auto-correlation values as described above.

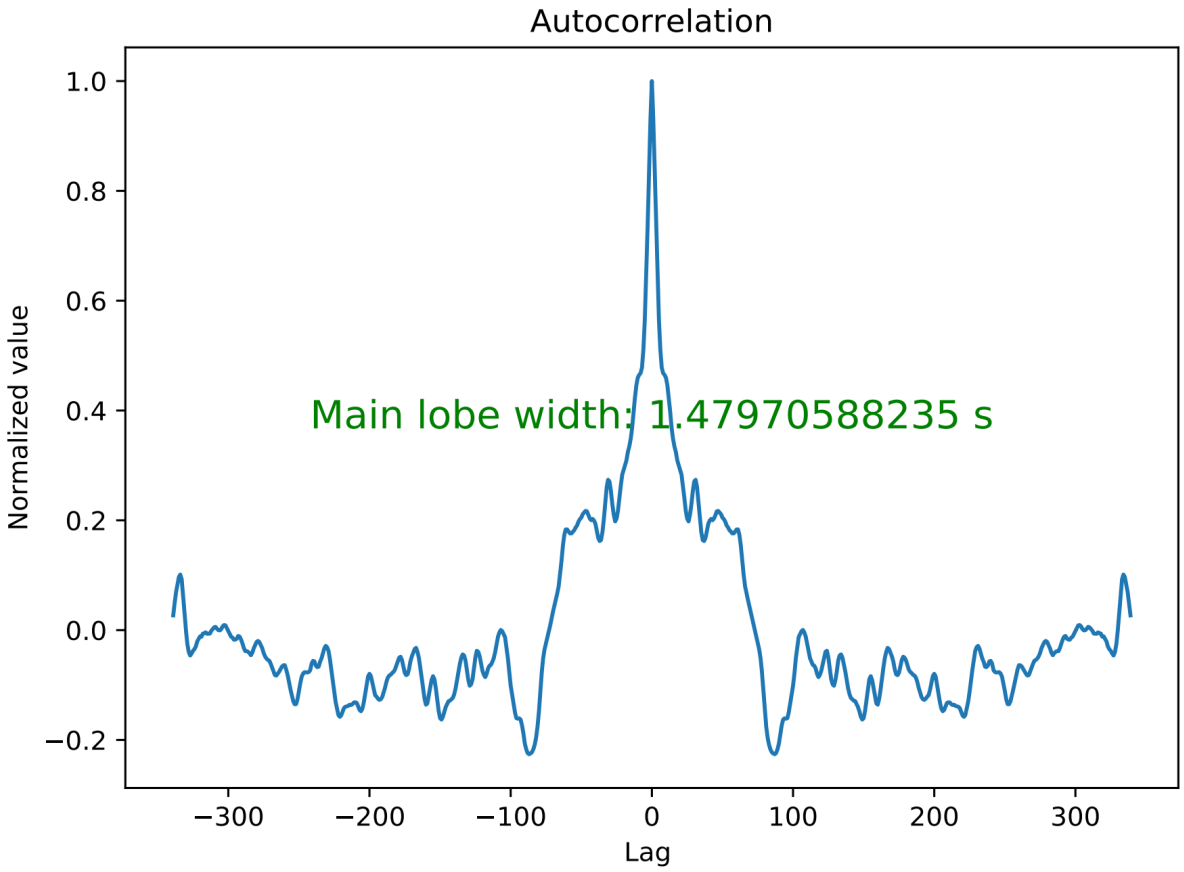


Figure 3.5: Autocorrelation function of a usual signal reflected from a human moving slightly

A flowchart showing the process to identify movements and to detect if a breathing-like signal is present in the radar return signal is shown in Figure 3.6.

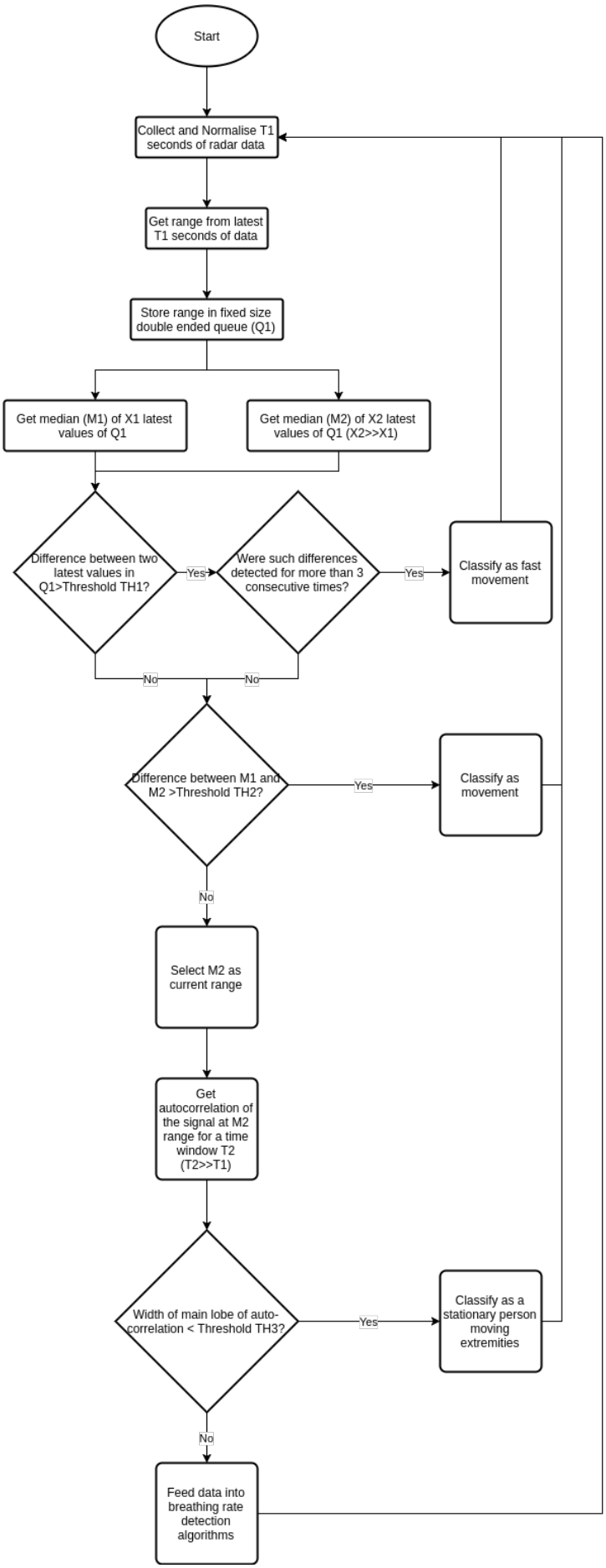


Figure 3.6: Flow chart of breathing signal detection algorithm

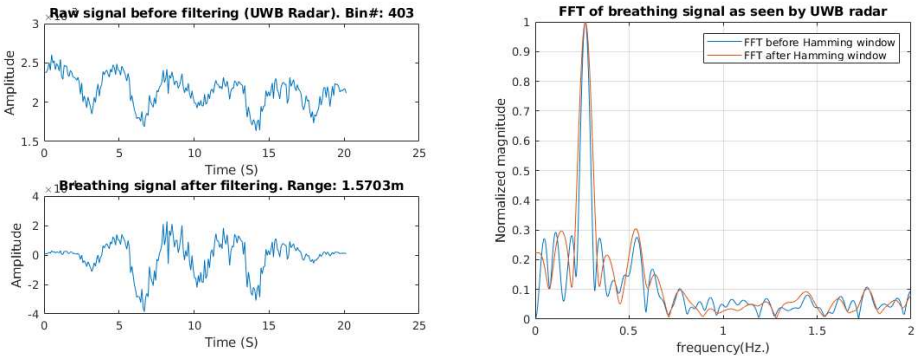
3.2 Breathing rate estimation

Performance of breathing rate estimation algorithms vary depending on many factors such as, breathing pattern of the person (eg: Shallow vs deep breathing, presence of sub-harmonics of the breathing rate, etc.), distance to the person from the radar, noise from body movements other than respiration etc. Therefore using just one algorithm in order to estimate breathing rate is not practical in real-world situations. A scoring system to evaluate the performance of algorithms in real-time without using any reference data is introduced in this research along with improved algorithms for breathing rate estimation.

3.2.1 FFT based breathing rate estimation

An FFT(Fast Fourier Transform) computes the Discrete Fourier Transform(DFT) and produces the exact same result as evaluating the DFT directly, albeit faster than the DFT. The breathing rate is continuously estimated in this work with overlapping windows of 15-20(adjustable) seconds with shifts of 1-3 (Upper limit adjustable) seconds between each consecutive window. The use of such narrow windows require the application of a proper windowing method such as applying a Hamming Window is required to prevent unwanted frequency components being added to the frequency spectra of the windowed signal. Presence of such unwanted components can be seen in Figure 3.7

The signal is also subjected to a band-pass filter to remove unwanted frequencies that do not lie within probable breathing rates (0.1Hz to 0.8Hz). Next, a Hampel filter, and a median



(a) Time domain signal (b) Frequency domain signal

Figure 3.7: Time and frequency domain breathing signals before and after applying Hamming window

filter is used to smoothen the signal. Next, the frequency corresponding to the maximum peak in amplitude in the frequency domain is selected as the breathing rate. As seen on Chapter 4, this method does not always work as the radar return signal from the chest does not always contain clear dominant peaks in the breathing frequency.

3.2.2 EMD based breathing rate estimation

The methods described in Section 2.3.3 lack the ability to select the correct IMF(s) which correspond to breathing automatically and even in cases where it is selected automatically, the results are not satisfactory as explained in Section 2.3.3 and further proven in Section 4.3. Therefore a novel method was developed with the ability to estimate breathing rate and extract breathing signal[2] automatically was developed.

One of the main considerations in developing this algorithm was the poor performance of other existing algorithms under real-world scenarios as it can be seen on the Chapter 4. The new algorithm was also developed with plans to launch this research work as a startup which caters to monitoring patients in elder-care facilities such as retirement homes and long term care facilities. Hence, the ease of implementing this algorithm on an embedded system such as a single board computer and the use of open-source libraries were another factor considered during the development. This algorithm was developed as part of the real-time processing system described in Section 3.6.

Luukko et al. in [35] introduces a C library with the ability to calculate the Empirical Mode Decomposition (EMD), Ensemble Empirical Mode Decomposition (EEMD) and Complete Ensemble Empirical Mode Decomposition (CEEMDAN). One of the major advantages of a low level C library for such a task is its ability to be compiled in almost any operating system and hardware configuration and this gives a system using such a library great flexibility in terms of hardware/software requirements. Luukko et al. has made their work open source under the GNU General Public license and they have also made wrappers for this library available in Python and R programming languages. The availability of this library in multiple languages and it's efficient implementation in C programming language makes it ideal for use in the applications described above.

[2] signal corresponding to the chest movement during breathing

EMD method experiences an issue named "mode mixing" which is the presence of oscillations of very disparate amplitude in a mode or the presence of very similar oscillations in different modes. Wu and Huang in [36] introduces an improved version of empirical mode decomposition called ensemble empirical mode decomposition (EEMD) to overcome these issues by performing EMD over an ensemble of the signal with added Gaussian white noise. The EEMD method performed on a signal $x[n]$ can be described as:

1. Generate $x^i[n] = x[n] + w^i[n]$, where $w^i[n](i = 1, 2, ..., I)$ are different realizations of white Gaussian noise, and I is the number of signals created by adding noise onto the original signal $x[n]$.

2. Each $x^i[n](i = 1, 2, ...I)$ is fully decomposed by EMD getting their nodes $IMF_k^i[n]$ where $k = 1, 2, ..., K$ indicates the modes.

3. Assign $\overline{IMF}_k$ as the k-th mode of $x[n]$ obtained as the average of the corresponding IMF_k^i. *i.e.* $\overline{IMF}_k[n] = \frac{1}{I}\sum_{i=1}^{I} IMF_k^i[n]$

The complete ensemble empirical mode decomposition introduced by Torres et al. in [37] is different from the EEMD method since it does not decompose each $x^i[n]$ independently from the other realizations. *i.e.* in CEEMDAN the residue $r_k^i[n]$ is calculated as $r_k^i[n] = r_{k-1}^i[n] - IMF_k^i[n]$. For example, in the method presented by Torres et al., the first residue is calculated as:

$$r_1[n] = x[n] - \widetilde{IMF}_1[n] \tag{3.1}$$

where $\widetilde{IMF}_1[n]$ is obtained in the same way as in EEMD. The complete procedure of CEEMDAN introduced by Torres et al. is described in the following paragraph.

Let $E_j(\cdot)$ be an operator, which, given a signal, produces the j^{th} mode obtained by EMD. Let w^ibe white noise with $\mathcal{N}(0, 1)$. If $x[n]$ is the targeted data, the CEEMDAN algorithm can be described as:

1. Decompose by EMD, I realizations $x[n]+\epsilon_0 w^i[n]$ to obtaind their first modes and compute;

$$\widetilde{IMF}_1[n] = \frac{1}{I}\sum_{i=1}^{I} IMF_1^i[n] = \overline{IMF_1}[n]$$

2. Calculate the first residue ($k = 1$) as $r_1[n] = x[n] - \widetilde{IMF}_1[n]$

3. Decompose realization $r_1[n] + \epsilon_1 E_1(w^i[n]), i = 1, 2, ..., I$, until their first EMD mode and define the second mode as:

$$\widetilde{IMF}_2[n] = \frac{1}{I}\sum_{i=1}^{I} E_1(r_1[n] + \epsilon_1 E_1(w^i[n]))$$

4. For $k = 2, ..., K$ calculate the k^{th} residue as:

$$r_k[n] = r_{(k-1)}[n] - \widetilde{IMF}_k[n]$$

5. Decompose realizations $r_k[n] + \epsilon_k E_k(w^i[n]), i = 1, 2, ..., I$, until their first EMD mode and define the $k + 1^{\text{th}}$ mode as:

$$\widetilde{IMF}_{(k+1)}[n] = \frac{1}{I}\sum_{i=1}^{I} E_1(r_k[n] + \epsilon_k E_k(w^i[n]))$$

6. Go to step 4 for next k

Steps 4 to 6 are performed until the obtained residue does not have at least two extrema. And the final residue satisfies $R[n] = x[n] - \sum_{k=1}^{K} \widetilde{IMF}_k$ with K being the total number of modes.

Even though the CEEMDAN method proposed by Torres et al. reduces the mode-mixing problem in EMD, when CEEMDAN was applied to breathing signals, the breathing signal seemed to be divided up into multiple modes in some cases specially when the monitored persons' inspiration and respiration follows different patterns/depths. Therefore the next step was to identify such situations and make sure the breathing signal is correctly captured by combining the multiple modes in such cases. To do this, the proposed algorithm checks if difference between the dominant rate of oscillation in each neighbouring mode is different less than δf Hz and if that is the case, the two neighbouring modes are added together to form one signal. This is done recursively for two levels, *i.e.* if the difference between the dominant rate of oscillation of $\widetilde{IMF}_k$ and $\widetilde{IMF}_{(k+1)}$ and the difference between the dominant rate of oscillation of $\widetilde{IMF}_{(k+1)}$ and $\widetilde{IMF}_{(k+2)}$ is less than δf, all three modes $k, k + 1$ and $k + 2$ are joined together. A value of 0.07 Hz for δf was found to be optimal after testing with breathing signals of several subjects. Let the set of signals obtained after combining similar $\widetilde{IMF}$s be $\widehat{IMF}_s$ where $s = 1, 2, ..., S$ with $S \leq K$

Next step is to select the correct $\widehat{IMF}$ corresponding to breathing of the monitored subject automatically and this is done by using a scoring algorithm that gives a score to each $\widehat{IMF}$ based on six factors.

1. **Ratio of energy contained within the $\widehat{IMF}$ vs original signal**: Let the total energy contained within the $\widehat{IMF}$ be $E_{\widehat{IMF}}$ and the total energy contained in the signal before CEEMDAN be E_{signal}. Then this ratio is defined as $\frac{E_{\widehat{IMF}}}{E_{signal}}$. Since the presence of a breathing-like signal highly likely at this stage because of the tests performed in Section 3.1 and hence, the $\widehat{IMF}$ corresponding to breathing should contain most of the power within the detected signal.

2. **Ratio of energy within two frequency ranges**: Within each individual $\widehat{IMF}$, the energy (E_1 and E_2) contained within two frequency ranges, (f_{1l} to f_{1h} and f_{2l} to f_{2h}) is calculated and the ratio between these two energy values can be expressed as: $\frac{E_1}{E_2}$. Optimal values used in the final system are: $f_{1l} = 0.17\text{Hz}, f_{1h} = 1\text{Hz}, f_{2l} = 1\text{Hz}, f_{2h} = 5\text{Hz}$

3. **Difference between the dominant rate of oscillation of original signal vs the $\widehat{IMF}$** : The absolute difference between the dominant rate of oscillation (Estimated by selecting the frequency corresponding to the maximum amplitude value of the discrete Fourier transform of the signal) of the original signal before CEEMDAN and the $\widehat{IMF}$.

4. **Correlation with DWT(Symlets6) approximation signal**: The correlation coefficient between the approximation signal obtained after performing discrete wavelet transform (DWT) on the original signal before CEEMDAN and the $\widehat{IMF}$. The 6$^{\text{th}}$ order Symlets wavelet is used for the Wavelet transform. The rationale behind using this parameter for scoring is built on Labarte et al.s work in [30] as described in Section 2.3.3.

5. **Difference between median of previous breathing rate estimates vs the estimate from $\widehat{IMF}$**: The breathing rate is estimated using the currently evaluated $\widehat{IMF}$ and the absolute difference of this value with the median of the previous estimates is used for scoring. The rationale behind this is the fact that the breathing rate does not change rapidly between estimates.

6. **Minkowski distance with the original signal**: The Minkowski distance between the original signal and the $\widehat{IMF}$ is defined by $d_{mink} = \left(\sum_{i=1}^{n} |sig_i - \widehat{IMF}_i|^2\right)^{1/2}$ where sig_i and $\widehat{IMF}_i$ are the i^{th} respective samples of the observed radar signal and the $\widehat{IMF}$ respectively and n is the number of samples in time domain. This parameter is used as Xinyang Zhang in [31] claims that the IMF corresponding to breathing rate should have the minimum Minkowski distance with the original signal (Section 2.3.3 states why this is not always true and is further emphasized in Section 4.3 where results using Xinyang Zhangs method is compared with other algorithms)

After calculating the six parameters above the different weights for the set of $\widehat{IMF}$s are normalized individually to values between -1 and $+1$. And then they are individually weighted to calculate a score for each $\widehat{IMF}$. The optimal weights identified during this research are (in the same order as the list above): 1.1, 0.2, -2, 0.5, -1, -1. These optimal values were identified by first making the weights of all parameters equal to 0 and slowly increasing/decreasing individual parameter weights and observing the effect on the mean error rate for multiple subjects for multiple test cases. The automated report generation system described in Section 3.5 was used for observing the effect of each parameter on multiple test cases. The $\widehat{IMF}$ with the maximum score after applying these weights are selected as the breathing signal in this algorithm. Next the dominant rate of oscillation of this $\widehat{IMF}$ is estimated by taking the discrete Fourier transform and selecting the frequency corresponding to the maximum amplitude in the frequency domain.

3.2.3 NMD based breathing rate estimation

Iatsenko et al. in [38] introduces a novel method to decompose a signal composed of a mixture of different oscillations, separating each oscillating component and background noise from each other. The signal is tracked through time using time frequency analysis (using windowed Fourier transform) and decomposed into a set of nonlinear modes [38]. The basic steps of this method are as follows;

1. Extract the fundamental harmonic of a nonlinear mode accurately from the signal's time frequency representation

2. Find candidates for all its possible harmonics

3. Identify the true harmonics (i.e. corresponding to the same nonlinear mode) among them

4. Subtract the resultant nonlinear mode from the signal and iterate the procedure on the residual until a preset stopping criterion is met

After extracting the signal corresponding to the fundamental harmonic, its fundamental frequency is calculated simply using the Fourier transform of the extracted signal (The frequency corresponding to the maximum amplitude in the frequency domain is obtained similar to the method described in Section 3.2.1). As seen on Section 4.3, the NMD based method is able to produce very good results.

3.2.4 Automatic selection of best performing algorithm

Since a single method can not produce good results in all situations, a method to evaluate the performance of the breathing rate estimation algorithms and select the best performing algorithm dynamically in real-time was developed. This system gives a score to breathing rate estimation algorithms through time and at each time instance, the breathing rate value from the algorithm with the highest score is selected as the current breathing rate. The scores are calculated using the following parameters:

1. **Majority voting**: A score is given to each algorithm based on how many of the other algorithms generate similar results. To allow for small differences in breathing rate within algorithms this score is calculated by creating a histogram by dividing up the range between 5 and 30 into bins of size 2 (all units are in breaths per minute). *i.e.* bin 1: 5 to 7, bin 2: 7 to 9, etc. Each breathing rate estimation from different algorithms are placed in these bins and the bin with the maximum number of algorithms is selected as the majority bin. All the algorithms belonging to the majority bin is given 2 points while the algorithms belonging to the two adjacent bins to the majority bin are given 1 point.

2. **Changes to breathing rate within the last 6 estimates**: A score is generated based on the absolute differences between the last six estimates. Let the last six breath rate estimates at time $t, t-1, ..., t-4, t-5$ be $\mathrm{BRate}(t), \mathrm{BRate}(t-1), ...\mathrm{BRate}(t-5)$ respectively. The parameter $\mathcal{CSCORE}$ defined as shown in Equation 3.2 is added to the score of each

algorithm.

$$\begin{aligned}\mathcal{CSCORE} = &-2.5\times \mid \mathrm{BRate}(t) - \mathrm{BRate}(t-1) \mid \\ &- 0.5\times \mid \mathrm{BRate}(t-1) - \mathrm{BRate}(t-2) \mid \\ &- 0.4\times \mid \mathrm{BRate}(t-2) - \mathrm{BRate}(t-3) \mid \\ &- 0.3\times \mid \mathrm{BRate}(t-3) - \mathrm{BRate}(t-4) \mid \\ &- 0.2\times \mid \mathrm{BRate}(t-4) - \mathrm{BRate}(t-5) \mid\end{aligned} \tag{3.2}$$

It should be noted that $\mathrm{BRate}(t), \mathrm{BRate}(t-1), ...$ in the above equation corresponds to the historical breathing rates obtained using the algorithm that is being evaluated (scored). *i.e.* $\mathrm{BRate}(t), ...$ is different for each algorithm that is being scored.

This system is capable of evaluating the performance of any number of algorithms at once and different algorithms can be included or excluded from the system with minimal modifications as it is built using high performance Python container data types[3] which are able to change their size/shape depending on the number of algorithms used. The optimal weights for the scoring algorithm was found by individually changing parameter weights and observing the results on the output using the automated report generation framework described in Section 3.5.

3.3 Data collection

The data collection system consists of a Matlab™ application that collects radar data (using a Xethru™ X4, or a Flatearth ancho™ radar) and simultaneously communicates with a C++ application that collects reference breathing data using a BioRadio™ 150 breathing belt sensor by GLneurotech®. Data was also collected using a linear actuator which was used to mimic periodic movements in order to test the radars ability to detect such periodic movements. An application provided by the linear actuator (Model: Actuonix® PQ-12P) manufacturer was modified in order to save the movements of the linear actuator (As discrete locations sampled at 20Hz) into a csv file for post processing. Figure 3.8 shows the user interface of the data collection application and a flowchart showing the operation of the data collection application is shown in 3.9.

[3] https://docs.python.org/2/library/collections.html

The data collection application starts up by reading a csv[4] file containing details about the test cases that needs to be run by the program. The details include a name for each test case, a short description and the duration for each individual test case. These details can be later edited through the user interface window before running the experiments. An example csv file demonstrating the format of the test case definition file is shown in Figure 3.10 and as seen in this figure, the first column contains a unique name for each test case and the second column contains small human readable description of each test case and the third column contains the duration of each test-case in seconds.

After the user interface is populated with the relevant data, the user selects the required settings and starts the data collection operation. Since multiple sensors need to collect data simultaneously, the Matlab® parallel computing toolbox[5] is used to create three extra individual processes which are responsible for capturing video, collecting radar data and a different process for synchronizing and communicating with other Matlab™ and C++ applications that are responsible for collecting data. The BioRadio™ 150 breathing belt sensor unit is controlled through a C++ based API provided by the manufacturer and hence, a simple C++ code was developed to connect to the breathing belt sensor and collect data. The data is then saved as csv files with relative timestamps received from the Matlab® socket server. The Actuonix® PQ-12P linear actuator mentioned above is also controlled using a similar C++ application when the data collection application is collecting data from a radar observing a moving linear actuator instead of a real human and the data is saved in a csv file. Due to the repeatability of linear actuator motions and the ability to manually synchronize the data collected using the linear actuator and the radar, the linear actuator application was developed with a much simpler method to synchronize the collected data from the radar and the linear actuator. The linear actuator controller application creates a new file with the current system time as its file name once every minute and saves the linear actuator positions in this file. These files are later read by a researcher who can manually use the correct file corresponding to the radar data. Figure 3.12 shows the synchronized signals captured using a breathing belt sensor and a radar and Figure 3.11 shows the linear actuator movements captured using the feedback sensor in the linear actuator and the movements seen by the radar.

[4]Comma separated values file

[5]https://www.mathworks.com/products/parallel-computing.html

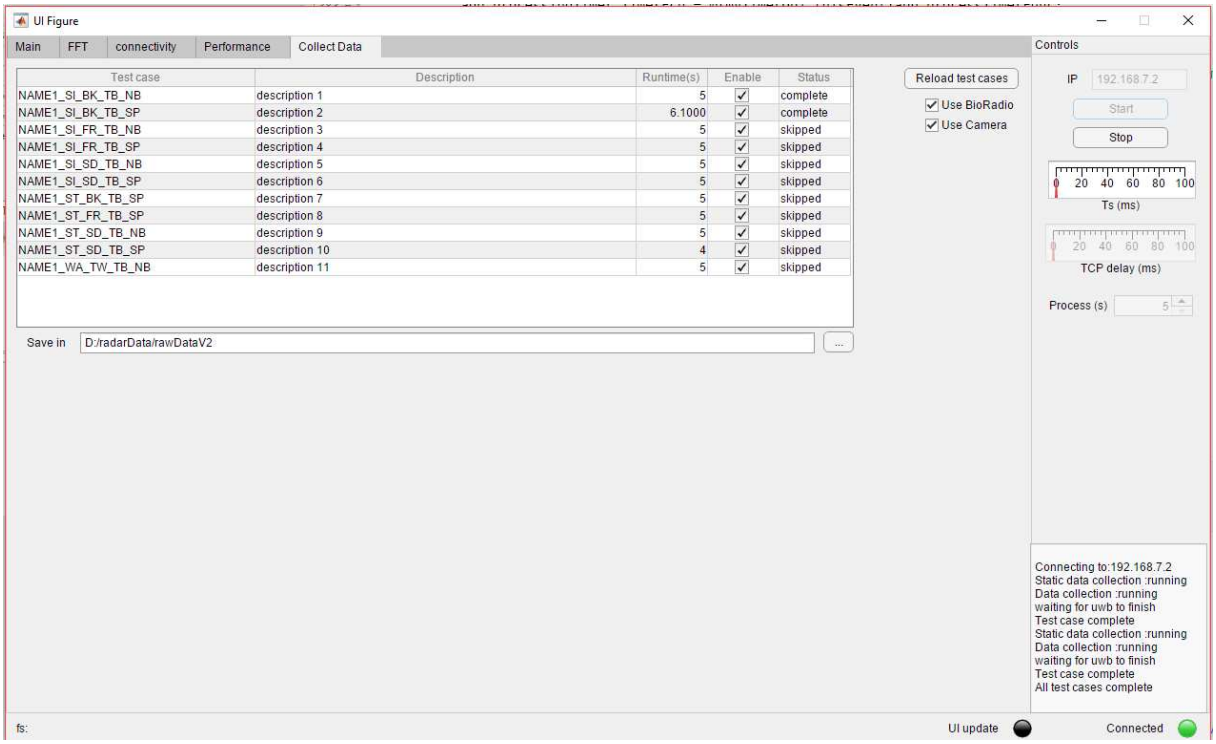


Figure 3.8: User interface of the data collection application

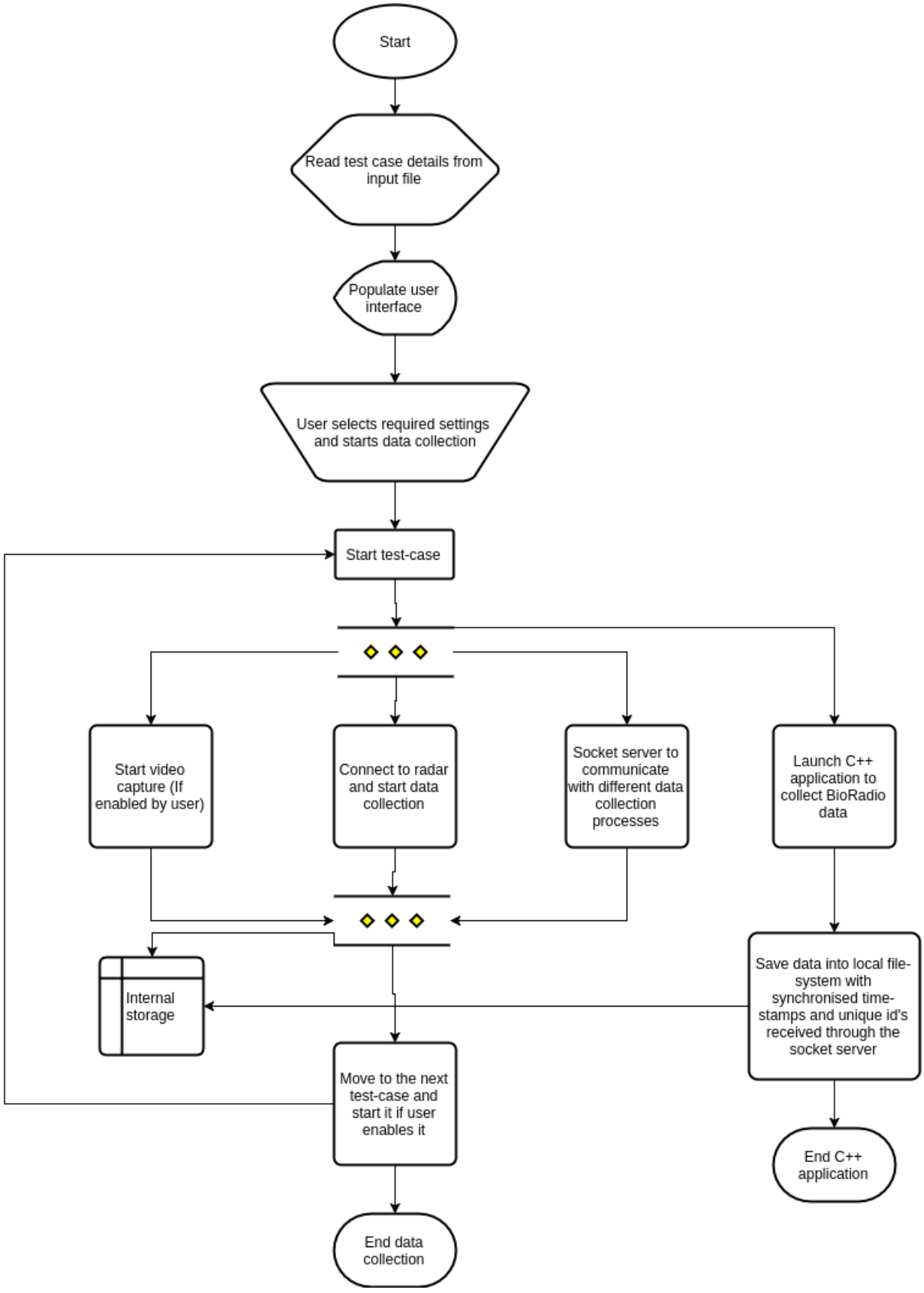


Figure 3.9: High level flowchart of the data collection application

ISURU-SIT-FR-P1	ISURU, sitting, front facing, point 1	60.5
ISURU-SIT-BA-P1	ISURU,SITTING,BACK FACING, POINT 1	124.25
ISURU-SIT-SL-P1	ISURU,SITTING, LEFT SIDE FACING, POINT 1	60
ISURU-STND-FR-P1	ISURU,STANDING,FRONT FACING, POINT1	60
ISURU-STND-BA-P1	ISURU,STANDING,BACK FACING, POINT1	60
ISURU-STND-SL-P1	ISURU,STANDING,LEFT SIDE FACING POINT 1	60
ISURU-SIT-FR-P2	ISURU, sitting, front facing, point 2	60
ISURU-SIT-BA-P2	ISURU,SITTING,BACK FACING, POINT 2	122.5
ISURU-SIT-SL-P2	ISURU,SITTING, LEFT SIDE FACING, POINT 2	60
ISURU-STND-FR-P2	ISURU,STANDING,FRONT FACING, POINT2	60
ISURU-STND-BA-P2	ISURU,STANDING,BACK FACING, POINT2	60
ISURU-STND-SL-P2	ISURU,STANDING,LEFT SIDE FACING POINT 2	60
ISURU-SIT-FR-P3	ISURU, sitting, front facing, point 3	60
ISURU-SIT-BA-P3	ISURU,SITTING,BACK FACING, POINT 3	60
ISURU-SIT-SL-P3	ISURU,SITTING, LEFT SIDE FACING, POINT 3	60
ISURU-STND-FR-P3	ISURU,STANDING,FRONT FACING, POINT3	60
ISURU-STND-BA-P3	ISURU,STANDING,BACK FACING, POINT3	60
ISURU-STND-SL-P3	ISURU,STANDING,LEFT SIDE FACING POINT 3	60
ISURU-SIT-FR-P4	ISURU, sitting, front facing, point 4	60
ISURU-SIT-BA-P4	ISURU,SITTING,BACK FACING, POINT 4	60
ISURU-SIT-SL-P4	ISURU,SITTING, LEFT SIDE FACING, POINT 4	60
ISURU-STND-FR-P4	ISURU,STANDING,FRONT FACING, POINT4	60
ISURU-STND-BA-P4	ISURU,STANDING,BACK FACING, POINT4	60
ISURU-STND-SL-P4	ISURU,STANDING,LEFT SIDE FACING POINT 4	60

Figure 3.10: Example csv file with multiple test-cases

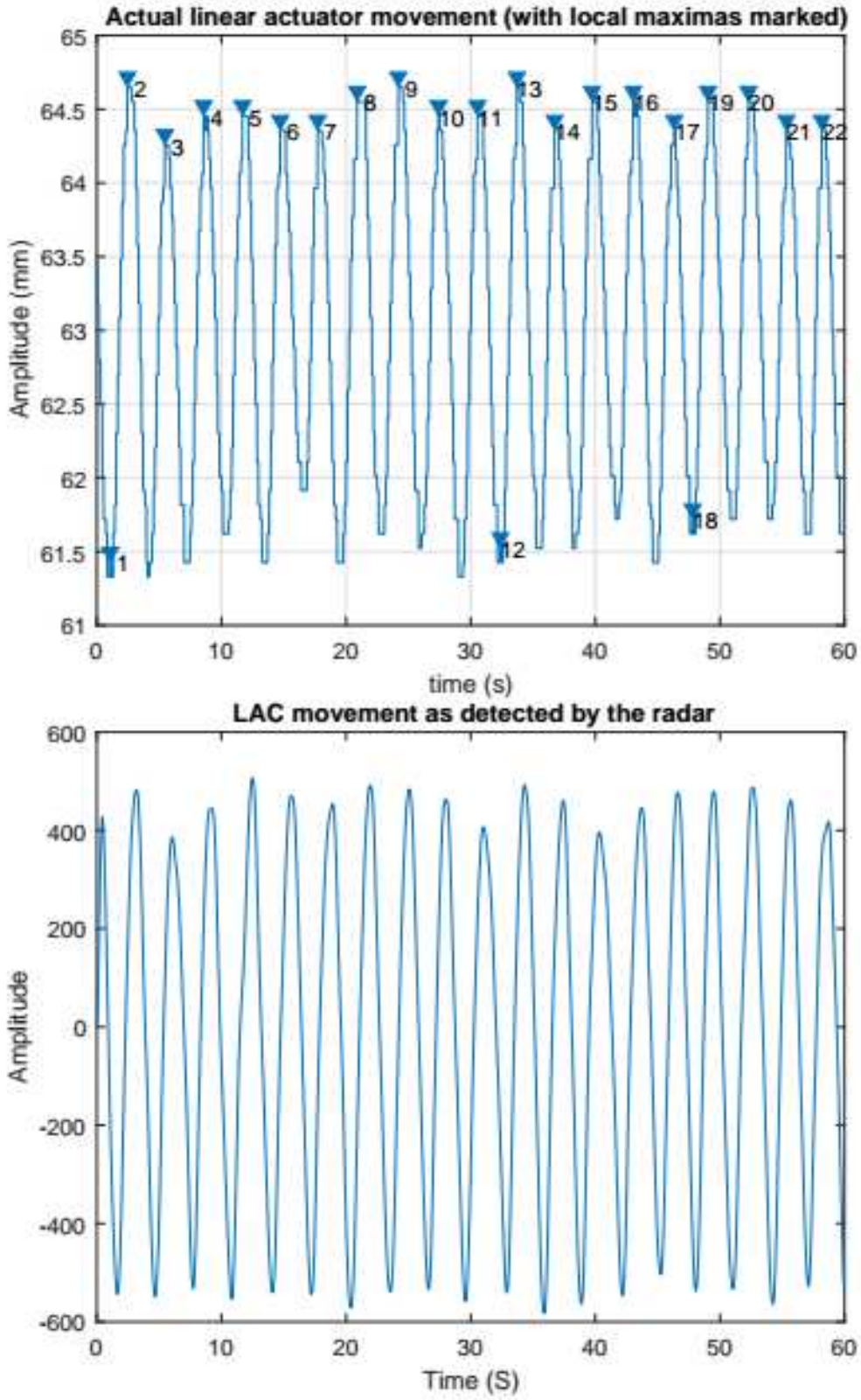
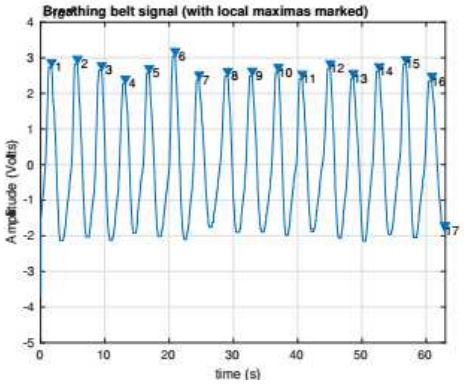
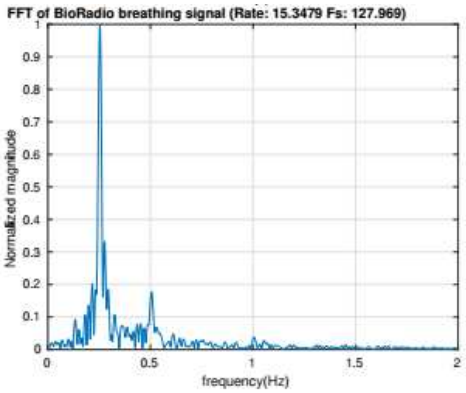


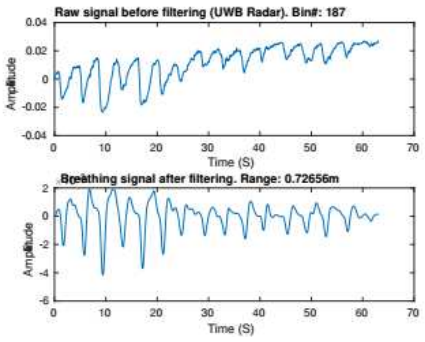
Figure 3.11: Linear actuator movements captured using the feedback sensor in the actuator and the radar



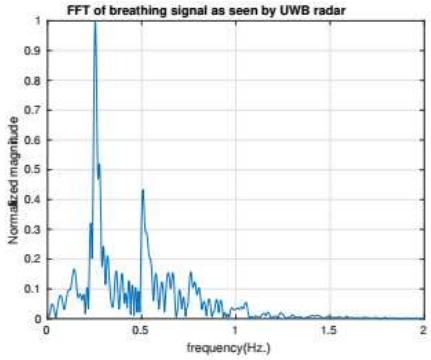
(a) Breath signal captured using breathing belt



(b) Discrete Fourier transform of breath signal captured using the breathing belt



(c) Breath signal captured using radar



(d) Discrete Fourier transform of breath signal captured using the radar

Figure 3.12: Breathing signal as seen by the radar and a breathing belt

3.4 Real-time breathing estimation on Matlab

Breathing estimation algorithms can only be truly beneficial if they are able to monitor a patient in real world situations such as in a nursing home/elder care facility etc. In order to use the algorithms to monitor people in such situations, the algorithms need to be executed on data as they are collected instead of processing them 'offline' after the data has been collected. Since most of the algorithms used in this research were written in Matlab®, a Matlab® based GUI application was developed in order to estimate breathing in real-time. The Matlab® Parallel computing toolbox was used to develop this application and different processes are created for collecting data from the radar, processing them, and displaying the results on a GUI. A screen capture of the Matlab® GUI can be found in Figure 3.13.

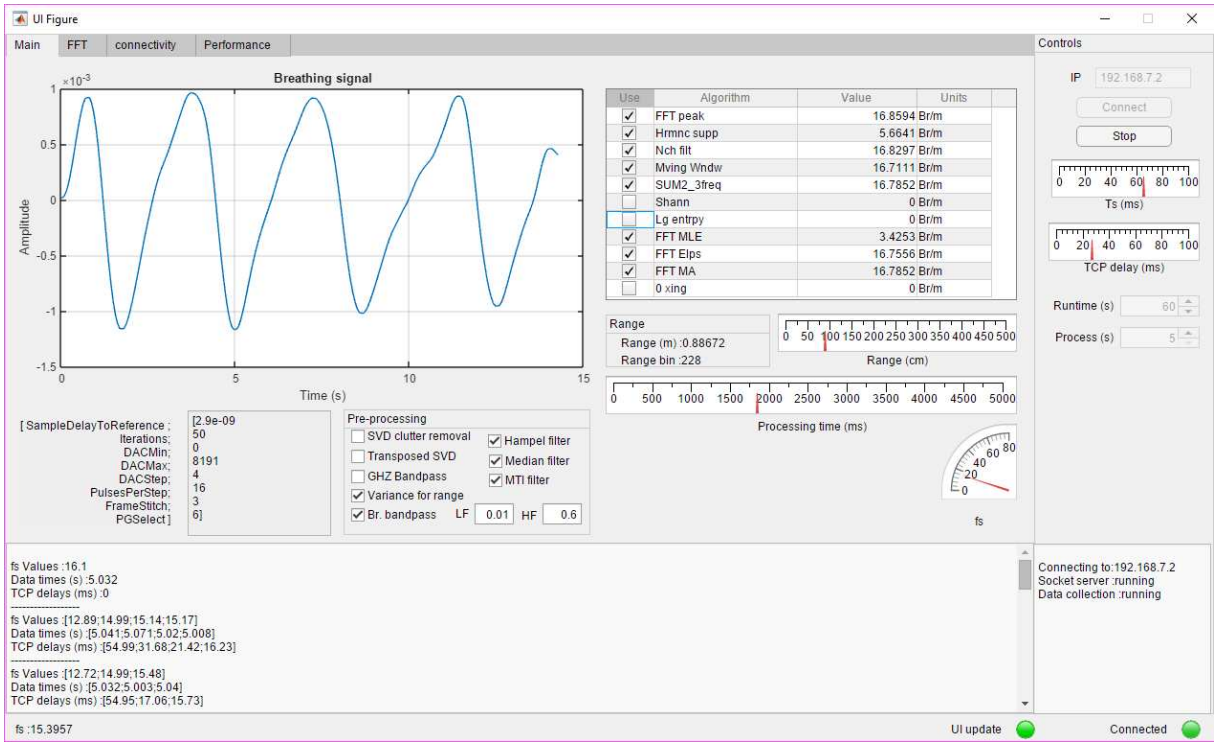


Figure 3.13: User interface of the Matlab application developed to estimate breathing rate

3.5 Algorithm analysis and report generation framework

A program was developed using Matlab® in order to efficiently analyze different algorithms and generate reports with their results. The pseudo-code describing the developed framework is shown in algorithm 3.1. As described in algorithm 3.1, the report generation program scans a given folder for all **.csv* data files and loads them up one by one. Then the data is divided into overlapping time windows as specified by the user and finds the breathing rate, breathing signal and the reference signal(from a linear actuator or breathing belt) and the reference rate corresponding to each time window using multiple algorithms. The results are returned from the *ANALYZE* function as two vectors which contain the names of the algorithms and the breathing rates calculated by those algorithms. These two vectors are then joined together as key:value pairs and saved on an Excel™ file which can be used as a summary of the performance of the algorithms. The generated Excel™ file contains information about the breathing rate estimates, error rates, mean error rate, correlation with the reference rate, for each time window corresponding to each algorithm. A LaTeXreport is also generated with details about all the important intermediary steps that the data goes through when they are analyzed by the algorithms. The report generation framework is developed in such a way so that the sizes of the *algorithm_names* and the *results* vectors are variable. This allows for the addition of new algorithms to reports just by adding new elements to the *algorithm_names* and *results* vectors. Thus adding new algorithms into the *ANALYZE* function will automatically expand the LaTeXreport and the auto-generated Excel™ file. Figure 3.14 shows an extract from an auto-generated detailed pdf report and Figure 3.15 shows an extract from an auto-generated Excel® sheet containing the summary of algorithm performance.

FFT_Ellipse fitting breathing rate	7.85 min^{-1}

Test case: ISUBU-STND-SL-P3

BioRadio Fs	127.97 *Hz*
Reference breathing rate (FFT belt)	14.18 min^{-1}

Sampling rate	12.13 Hz
Target range bin	302
Estimated distance	1.18 m
Breathing rate from Welch PSD	22.76 min^{-1}
Breathing rate from max FFT	22.78 min^{-1}
Breathing rate after harmonic suppression	7.55 min^{-1}
Notch filter based breathing rate	7.58 min^{-1}
Energy sum window based breathing rate	7.57 min^{-1}
Energy sum squared algorithm based breathing rate	7.57 min^{-1}
Shannon energy based algorithm breathing rate	7.57 min^{-1}
Log entropy based algorithm breathing rate	6.01 min^{-1}
FFT_MLE breathing rate	7.57 min^{-1}
FFT_Ellipse fitting breathing rate	7.60 min^{-1}

11

FFT_Ellipse fitting breathing rate	11.39 min^{-1}

Test case: ISUBU-SIT-SA-P4

BioRadio Fs	127.94 *Hz*
Reference breathing rate (FFT belt)	13.94 min^{-1}

Sampling rate	12.05 Hz
Target range bin	400
Estimated distance	1.59 m
Breathing rate from Welch PSD	16.94 min^{-1}
Breathing rate from max FFT	20.63 min^{-1}
Breathing rate after harmonic suppression	6.92 min^{-1}
Notch filter based breathing rate	13.24 min^{-1}
Energy sum window based breathing rate	6.91 min^{-1}
Energy sum squared algorithm based breathing rate	13.24 min^{-1}
Shannon energy based algorithm breathing rate	13.24 min^{-1}
Log entropy based algorithm breathing rate	173.05 min^{-1}
FFT_MLE breathing rate	6.92 min^{-1}
FFT_Ellipse fitting breathing rate	13.23 min^{-1}

12

Test case: ISUBU-SIT-SL-P4

BioRadio Fs	128.16 *Hz*
Reference breathing rate (FFT belt)	13.49 min^{-1}

Sampling rate	12.00 Hz
Target range bin	546
Estimated distance	2.13 m
Breathing rate from Welch PSD	22.50 min^{-1}
Breathing rate from max FFT	18.96 min^{-1}
Breathing rate after harmonic suppression	7.18 min^{-1}
Notch filter based breathing rate	9.56 min^{-1}
Energy sum window based breathing rate	14.42 min^{-1}
Energy sum squared algorithm based breathing rate	9.55 min^{-1}
Shannon energy based algorithm breathing rate	9.55 min^{-1}
Log entropy based algorithm breathing rate	6.01 min^{-1}
FFT_MLE breathing rate	6.32 min^{-1}
FFT_Ellipse fitting breathing rate	6.32 min^{-1}

Test case: ISUBU-STND-SA-P4

BioRadio Fs	127.87 *Hz*
Reference breathing rate (FFT belt)	14.75 min^{-1}

Sampling rate	11.93 Hz
Target range bin	540
Estimated distance	2.11 m
Breathing rate from Welch PSD	13.98 min^{-1}
Breathing rate from max FFT	14.59 min^{-1}
Breathing rate after harmonic suppression	7.38 min^{-1}
Notch filter based breathing rate	14.67 min^{-1}
Energy sum window based breathing rate	14.67 min^{-1}
Energy sum squared algorithm based breathing rate	14.64 min^{-1}
Shannon energy based algorithm breathing rate	14.69 min^{-1}
Log entropy based algorithm breathing rate	6.01 min^{-1}
FFT_MLE breathing rate	4.89 min^{-1}
FFT_Ellipse fitting breathing rate	14.70 min^{-1}

13

Test case: ISUBU-STND-FB-P4

BioRadio Fs	127.90 *Hz*
Reference breathing rate (FFT belt)	13.57 min^{-1}

Sampling rate	11.93 Hz
Target range bin	551
Estimated distance	2.15 m
Breathing rate from Welch PSD	13.98 min^{-1}
Breathing rate from max FFT	13.47 min^{-1}
Breathing rate after harmonic suppression	4.51 min^{-1}
Notch filter based breathing rate	6.90 min^{-1}
Energy sum window based breathing rate	6.90 min^{-1}
Energy sum squared algorithm based breathing rate	6.88 min^{-1}
Shannon energy based algorithm breathing rate	6.89 min^{-1}
Log entropy based algorithm breathing rate	6.01 min^{-1}
FFT_MLE breathing rate	6.90 min^{-1}
FFT_Ellipse fitting breathing rate	5.25 min^{-1}

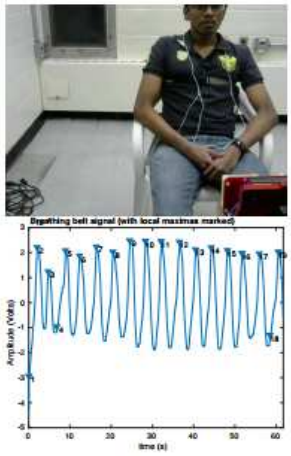
14

Test case: ISUBU-STND-SL-P4

BioRadio Fs	128.06 *Hz*
Reference breathing rate (FFT belt)	15.05 min^{-1}

Sampling rate	11.92 Hz
Target range bin	432
Estimated distance	1.68 m
Breathing rate from Welch PSD	16.76 min^{-1}
Breathing rate from max FFT	15.84 min^{-1}
Breathing rate after harmonic suppression	7.92 min^{-1}
Notch filter based breathing rate	15.83 min^{-1}
Energy sum window based breathing rate	15.83 min^{-1}
Energy sum squared algorithm based breathing rate	15.82 min^{-1}
Shannon energy based algorithm breathing rate	15.82 min^{-1}
Log entropy based algorithm breathing rate	6.01 min^{-1}
FFT_MLE breathing rate	5.27 min^{-1}
FFT_Ellipse fitting breathing rate	15.84 min^{-1}

15



16

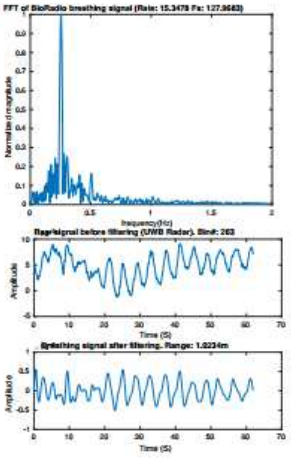
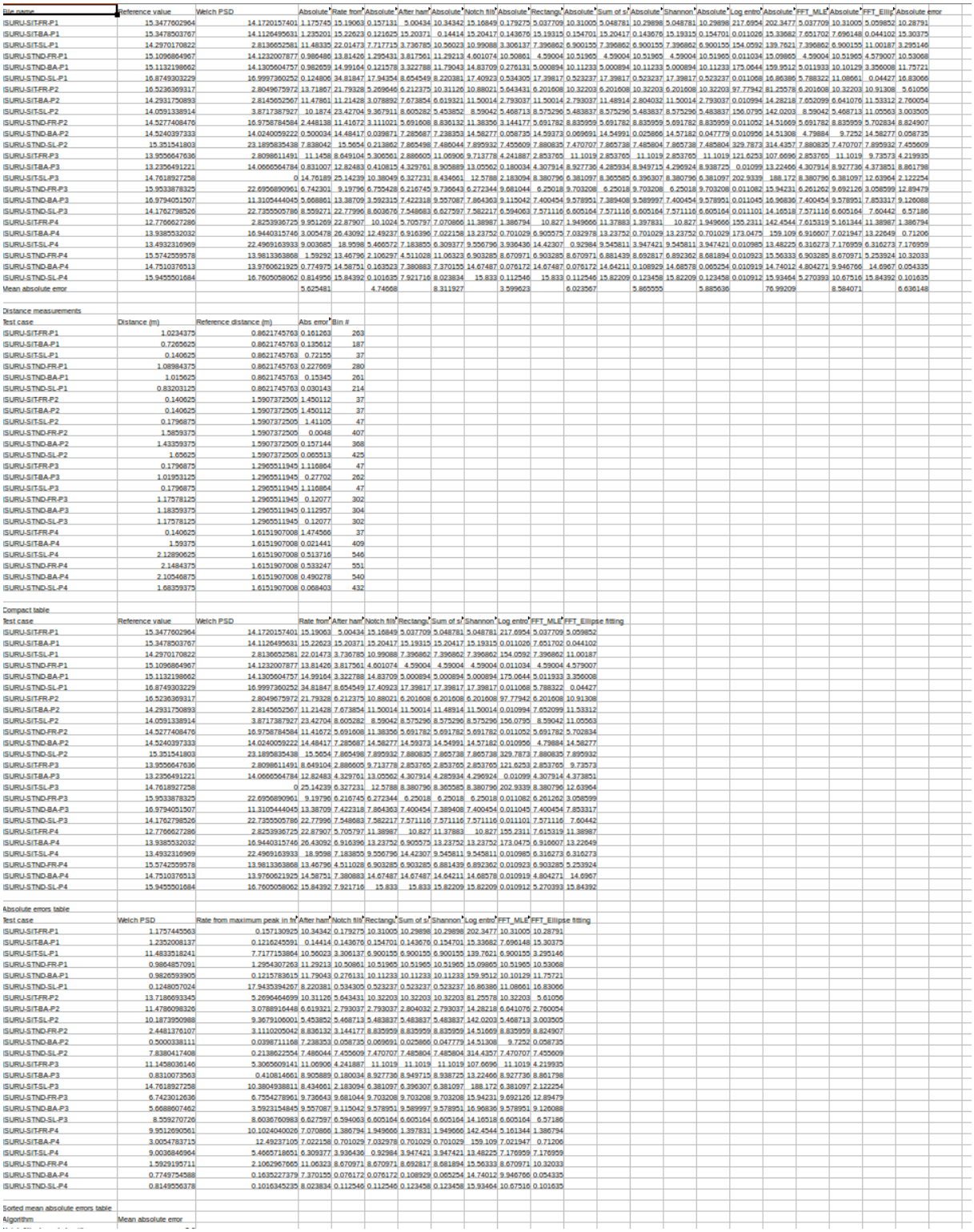


Figure 3.14: Extract from a detailed pdf report generated automatically using the report generation framework



File name	Reference value	Welch PSD	Absolute	Rate from	Absolute	After ham	Absolute	Notch filt	Absolute	Rectangl	Absolute	Sum of s	Absolute	Shannon	Absolute	Log entro	Absolute	FFT_MLE	Absolute	FFT_Ellip	Absolute error
ISURU-S/T-FR-P1	15.3477602964	14.1720157401	1.175745	15.19063	0.157131	5.00434	10.34342	15.16849	0.179275	5.037709	10.31005	5.048781	10.29898	5.048781	10.29898	217.6954	202.3477	5.037709	10.31005	5.059852	10.28791
ISURU-S/T-BA-P1	15.3478503767	14.1126495631	1.235201	15.22623	0.121625	15.20371	0.14414	15.20417	0.143676	15.19315	0.154701	15.20417	0.143676	15.19315	0.154701	0.011026	15.33682	7.651702	7.696148	0.044102	15.30375
ISURU-S/T-SL-P1	14.2970170822	2.8136652581	11.48335	22.01473	7.717115	3.736785	10.56023	10.99088	3.306137	7.396862	6.900155	7.396862	6.900155	7.396862	6.900155	154.0592	139.7621	7.396862	6.900155	11.00187	3.295146
ISURU-S/T-STND-FR-P1	15.1096864967	14.1232007877	0.986486	13.81426	1.295431	3.817561	11.29213	4.601074	10.50861	4.59004	10.51965	4.59004	10.51965	4.59004	10.51965	0.011034	15.09865	4.59004	10.51965	4.579007	10.53068
ISURU-S/T-STND-BA-P1	15.1132198662	14.1305604757	0.982659	14.99164	0.121578	3.322788	11.79043	14.837095	0.276131	5.000894	10.11233	5.000894	10.11233	5.000894	10.11233	175.0644	159.9512	5.011933	10.10129	3.356008	11.75721
ISURU-S/T-STND-SL-P1	16.8749303229	16.9997360252	0.124806	34.81847	17.94354	8.654549	8.220381	17.40923	0.534305	17.39817	0.523237	17.39817	0.523237	17.39817	0.523237	0.011068	16.86386	5.788322	11.08661	0.04427	16.83066
ISURU-S/T-FR-P2	16.5236369317	2.8049675972	13.71867	21.79328	5.269646	6.212375	10.31126	10.880215	5.643431	6.201608	10.32203	6.201608	10.32203	6.201608	10.32203	97.77942	81.25578	6.201608	10.32203	10.91308	5.61056
ISURU-S/T-BA-P2	14.2931750893	2.8145652567	11.47861	11.21428	3.078892	7.673854	6.619321	11.50014	2.793037	11.50014	2.793037	11.48914	2.804032	11.50014	2.793037	0.010994	14.28218	7.652099	6.641076	11.53312	2.760054
ISURU-S/T-SL-P2	14.0591338914	3.8717387927	10.1874	23.42704	9.367911	8.605282	5.453852	8.59042	5.468713	8.575296	5.483837	8.575296	5.483837	8.575296	5.483837	156.0795	142.0203	8.59042	5.468713	11.05563	3.003505
ISURU-S/T-STND-FR-P2	14.5277408476	16.9758784584	2.448138	11.41672	3.111021	5.691608	8.836132	11.38356	3.144177	5.691782	8.835959	5.691782	8.835959	5.691782	8.835959	0.011052	14.51669	5.691782	8.835959	5.702834	8.824907
ISURU-S/T-STND-BA-P2	14.5240397333	14.0240059222	0.500034	14.48417	0.039871	7.285687	7.238353	14.58277	0.058735	14.59373	0.069691	14.54991	0.025866	14.57182	0.047779	0.010956	14.51308	4.79884	9.7252	14.58277	0.058735
ISURU-S/T-STND-SL-P2	15.351541803	23.1895835438	7.838042	15.5654	0.213862	7.865498	7.486044	7.895932	7.455609	7.880835	7.470707	7.865738	7.485804	7.865738	7.485804	329.7873	314.4357	7.880835	7.470707	7.895932	7.455609
ISURU-S/T-FR-P3	13.9556647636	2.8098611491	11.1458	8.649104	5.306561	2.886605	11.06906	9.713778	4.241887	2.853765	11.1019	2.853765	11.1019	2.853765	11.1019	121.6253	107.6696	2.853765	11.1019	9.73573	4.219935
ISURU-S/T-BA-P3	13.2356491221	14.0666564784	0.831007	12.82483	0.410815	4.329761	8.905889	13.05562	0.180034	4.307914	8.927736	4.285934	8.949715	4.296924	8.938725	0.01099	13.22466	4.307914	8.927736	4.373851	8.861798
ISURU-S/T-SL-P3	14.7618927258	0	14.76189	25.14239	10.38049	6.327231	8.434661	12.5788	2.183094	8.380796	6.381097	8.365585	6.396307	8.381097	6.381097	202.9339	188.172	8.380796	6.381097	12.63964	2.122254
ISURU-S/T-STND-FR-P3	15.9533878325	22.6956890961	6.742301	9.19796	6.755428	6.216745	9.736643	6.272344	9.681044	6.25018	9.703208	6.25018	9.703208	6.25018	9.703208	0.011082	15.94231	6.261262	9.692126	3.058599	12.89479
ISURU-S/T-STND-BA-P3	16.9794051507	11.3105444045	5.668861	13.38709	3.592315	7.422318	9.557087	7.864363	9.115042	7.400454	9.578951	7.389408	9.589997	7.400454	9.578951	0.011045	16.96836	7.400454	9.578951	7.853317	9.126088
ISURU-S/T-STND-SL-P3	14.1762798526	22.7355505786	8.559271	22.77996	8.603676	7.548683	6.627597	7.582217	6.594063	7.571116	6.605164	7.571116	6.605164	7.571116	6.605164	0.011101	14.16518	7.571116	6.605164	7.60442	6.57186
ISURU-S/T-FR-P4	12.7766627286	2.8253936725	9.951209	22.87907	10.1024	5.705797	7.070866	11.38987	1.386794	10.827	1.949666	11.37883	1.397831	10.827	1.949666	155.2311	142.4544	7.615319	5.161344	11.38987	1.386794
ISURU-S/T-BA-P4	13.9385532032	16.9440315746	3.005478	26.43092	12.49237	6.916396	7.022158	13.23752	0.701029	6.905575	7.032978	13.23752	0.701029	13.23752	0.701029	173.0475	159.109	6.916607	7.021947	13.22649	0.71206
ISURU-S/T-SL-P4	13.4932316969	22.4969163933	9.003685	18.9598	5.466572	7.183855	6.309377	9.556796	3.936436	14.42307	0.92984	9.545811	3.947421	9.545811	3.947421	0.010985	13.48225	6.316273	7.176959	6.316273	7.176959
ISURU-S/T-STND-FR-P4	15.5742559578	13.9813363868	1.59292	13.46796	2.106297	4.511028	11.06323	6.903285	8.670971	6.903285	8.670971	6.881439	8.692817	6.892362	8.681894	0.010923	15.56333	6.903285	8.670971	5.253924	10.32033
ISURU-S/T-STND-BA-P4	14.7510376513	13.9760621925	0.774975	14.58751	0.163523	7.380883	7.370155	14.67487	0.076172	14.67487	0.076172	14.64211	0.108929	14.68578	0.065254	0.010919	14.74012	4.804271	9.946766	14.6967	0.054335
ISURU-S/T-STND-SL-P4	15.9455501684	16.7605058062	0.814956	15.84392	0.101635	7.921716	8.023834	15.833	0.112546	15.833	0.112546	15.82209	0.123458	15.82209	0.123458	0.010912	15.93464	5.270393	10.67516	15.84392	0.101635
Mean absolute error			5.625481		4.74668		8.311927		3.599623		6.023567		5.865555		5.885636		76.99209		8.584071		6.636148
Distance measurements																					
Test case	Distance (m)	Reference distance (m)	Abs error	Bin #																	
ISURU-S/T-FR-P1	1.0234375	0.8621745763	0.161263	263																	
ISURU-S/T-BA-P1	0.7265625	0.8621745763	0.135612	187																	
ISURU-S/T-SL-P1	0.140625	0.8621745763	0.72155	37																	
ISURU-S/T-STND-FR-P1	1.08984375	0.8621745763	0.227609	280																	
ISURU-S/T-STND-BA-P1	1.015625	0.8621745763	0.15345	261																	
ISURU-S/T-STND-SL-P1	0.83203125	0.8621745763	0.030143	214																	
ISURU-S/T-FR-P2	0.140625	1.5907372505	1.450112	37																	
ISURU-S/T-BA-P2	0.140625	1.5907372505	1.450112	37																	
ISURU-S/T-SL-P2	0.1796875	1.5907372505	1.41105	47																	
ISURU-S/T-STND-FR-P2	1.5859375	1.5907372505	0.0048	407																	
ISURU-S/T-STND-BA-P2	1.43359375	1.5907372505	0.157144	368																	
ISURU-S/T-STND-SL-P2	1.65625	1.5907372505	0.065513	425																	
ISURU-S/T-FR-P3	0.1796875	1.2965511945	1.116864	47																	
ISURU-S/T-BA-P3	0.01953125	1.2965511945	0.27702	262																	
ISURU-S/T-SL-P3	0.1796875	1.2965511945	1.116864	47																	
ISURU-S/T-STND-FR-P3	1.17578125	1.2965511945	0.12077	302																	
ISURU-S/T-STND-BA-P3	1.18359375	1.2965511945	0.112957	304																	
ISURU-S/T-STND-SL-P3	1.17578125	1.2965511945	0.12077	302																	
ISURU-S/T-FR-P4	0.140625	1.6151907008	1.474566	37																	
ISURU-S/T-BA-P4	1.59375	1.6151907008	0.021441	409																	
ISURU-S/T-SL-P4	2.12890625	1.6151907008	0.513716	546																	
ISURU-S/T-STND-FR-P4	2.1484375	1.6151907008	0.533247	551																	
ISURU-S/T-STND-BA-P4	2.10546875	1.6151907008	0.490278	540																	
ISURU-S/T-STND-SL-P4	1.68359375	1.6151907008	0.068403	432																	
Compact table																					
Test case	Reference value	Welch PSD	Rate from	After ham	Notch filt	Rectangl	Sum of s	Shannon	Log entro	FFT_MLE	FFT_Ellipse fitting										
ISURU-S/T-FR-P1	15.3477602964	14.1720157401	15.19063	5.00434	15.16849	5.037709	5.048781	5.048781	217.6954	5.037709	5.059852										
ISURU-S/T-BA-P1	15.3478503767	14.1126495631	15.22623	15.20371	15.20417	15.19315	15.20417	15.19315	0.011026	7.651702	0.044102										
ISURU-S/T-SL-P1	14.2970170822	2.8136652581	22.01473	3.736785	10.99088	7.396862	7.396862	7.396862	154.0592	7.396862	11.00187										
ISURU-S/T-STND-FR-P1	15.1096864967	14.1232007877	13.81426	3.817561	4.601074	4.59004	4.59004	4.59004	0.011034	4.59004	4.579007										
ISURU-S/T-STND-BA-P1	15.1132198662	14.1305604757	14.99164	3.322788	14.83709	5.000894	5.000894	5.000894	175.0644	5.011933	3.356008										
ISURU-S/T-STND-SL-P1	16.8749303229	16.9997360252	34.81847	8.654549	17.40923	17.39817	17.39817	17.39817	0.011068	5.788322	0.04427										
ISURU-S/T-FR-P2	16.5236369317	2.8049675972	21.79328	6.212375	10.88021	6.201608	6.201608	6.201608	97.77942	6.201608	10.91308										
ISURU-S/T-BA-P2	14.2931750893	2.8145652567	11.21428	7.673854	11.50014	11.50014	11.48914	11.50014	0.010994	7.652099	11.53312										
ISURU-S/T-SL-P2	14.0591338914	3.8717387927	23.42704	8.605282	8.59042	8.575296	8.575296	8.575296	156.0795	8.59042	11.05563										
ISURU-S/T-STND-FR-P2	14.5277408476	16.9758784584	11.41672	5.691608	11.38356	5.691782	5.691782	5.691782	0.011052	5.691782	5.702834										
ISURU-S/T-STND-BA-P2	14.5240397333	14.0240059222	14.48417	7.285687	14.58277	14.59373	14.54991	14.57182	0.010956	4.79884	14.58277										
ISURU-S/T-STND-SL-P2	15.351541803	23.1895835438	15.5654	7.865498	7.895932	7.880835	7.865738	7.865738	329.7873	7.880835	7.895932										
ISURU-S/T-FR-P3	13.9556647636	2.8098611491	8.649104	2.886605	9.713778	2.853765	2.853765	2.853765	121.6253	2.853765	9.73573										
ISURU-S/T-BA-P3	13.2356491221	14.0666564784	12.82483	4.329761	13.05562	4.307914	4.285934	4.296924	0.01099	4.307914	4.373851										
ISURU-S/T-SL-P3	14.7618927258	0	25.14239	6.327231	12.5788	8.380796	8.365585	8.380796	202.9339	8.380796	12.63964										
ISURU-S/T-STND-FR-P3	15.9533878325	22.6956890961	9.19796	6.216745	6.272344	6.25018	6.25018	6.25018	0.011082	6.261262	3.058599										
ISURU-S/T-STND-BA-P3	16.9794051507	11.3105444045	13.38709	7.422318	7.864363	7.400454	7.389408	7.400454	0.011045	7.400454	7.853317										
ISURU-S/T-STND-SL-P3	14.1762798526	22.7355505786	22.77996	7.548683	7.582217	7.571116	7.571116	7.571116	0.011101	7.571116	7.60442										
ISURU-S/T-FR-P4	12.7766627286	2.8253936725	22.87907	5.705797	11.38987	10.827	11.37883	10.827	155.2311	7.615319	11.38987										
ISURU-S/T-BA-P4	13.9385532032	16.9440315746	26.43092	6.916396	13.23752	6.905575	13.23752	13.23752	173.0475	6.916607	13.22649										
ISURU-S/T-SL-P4	13.4932316969	22.4969163933	18.9598	7.183855	9.556796	14.42307	9.545811	9.545811	0.010985	6.316273	6.316273										
ISURU-S/T-STND-FR-P4	15.5742559578	13.9813363868	13.46796	4.511028	6.903285	6.903285	6.881439	6.892362	0.010923	6.903285	5.253924										
ISURU-S/T-STND-BA-P4	14.7510376513	13.9760621925	14.58751	7.380883	14.67487	14.67487	14.64211	14.68578	0.010919	4.804271	14.6967										
ISURU-S/T-STND-SL-P4	15.9455501684	16.7605058062	15.84392	7.921716	15.833	15.833	15.82209	15.82209	0.010912	5.270393	15.84392										
Absolute errors table																					
Test case	Welch PSD	Rate from maximum peak in fr	After ham	Notch filt	Rectangl	Sum of s	Shannon	Log entro	FFT_MLE	FFT_Ellipse fitting											
ISURU-S/T-FR-P1	1.1757445563	0.157130925	10.34342	0.179275	10.31005	10.29898	10.29898	202.3477	10.31005	10.28791											
ISURU-S/T-BA-P1	1.2352008137	0.1216245591	0.14414	0.143676	0.154701	0.143676	0.154701	15.33682	7.696148	15.30375											
ISURU-S/T-SL-P1	11.4833518241	7.7177153864	10.56023	3.306137	6.900155	6.900155	6.900155	139.7621	6.900155	3.295146											
ISURU-S/T-STND-FR-P1	0.9864857091	1.2954307263	11.29213	10.50861	10.51965	10.51965	10.51965	15.09865	10.51965	10.53068											
ISURU-S/T-STND-BA-P1	0.9826593905	0.1215783615	11.79043	0.276131	10.11233	10.11233	10.11233	159.9512	10.10129	11.75721											
ISURU-S/T-STND-SL-P1	0.1248057024	17.9435394267	8.220381	0.534305	0.523237	0.523237	0.523237	16.86386	11.08661	16.83066											
ISURU-S/T-FR-P2	13.7186693345	5.2696464699	10.31126	5.643431	10.32203	10.32203	10.32203	81.25578	10.32203	5.61056											
ISURU-S/T-BA-P2	11.4786098326	3.0788916448	6.619321	2.793037	2.793037	2.804032	2.793037	14.28218	6.641076	2.760054											
ISURU-S/T-SL-P2	10.1873950988	9.3671906001	5.453852	5.468713	5.483837	5.483837	5.483837	142.0203	5.468713	3.003505											
ISURU-S/T-STND-FR-P2	2.4481376107	3.1110205042	8.836132	3.144177	8.835959	8.835959	8.835959	14.51669	8.835959	8.824907											
ISURU-S/T-STND-BA-P2	0.5000338111	0.0398711168	7.238353	0.058735	0.069691	0.025866	0.047779	14.51308	9.7252	0.058735											
ISURU-S/T-STND-SL-P2	7.8380417408	0.2138622554	7.486044	7.455609	7.470707	7.485804	7.485804	314.4357	7.470707	7.455609											
ISURU-S/T-FR-P3	11.1458036146	5.3065609141	11.06906	4.241887	11.1019	11.1019	11.1019	107.6696	11.1019	4.219935											
ISURU-S/T-BA-P3	0.8310073563	0.4108146661	8.905889	0.180034	8.927736	8.949715	8.938725	13.22466	8.927736	8.861798											
ISURU-S/T-SL-P3	14.7618927258	10.3804938811	8.434661	2.183094	6.381097	6.396307	6.381097	188.172	6.381097	2.122254											
ISURU-S/T-STND-FR-P3	6.7423012636	6.7554278961	9.736643	9.681044	9.703208	9.703208	9.703208	15.94231	9.692126	12.89479											
ISURU-S/T-STND-BA-P3	5.6688607462	3.5923154845	9.557087	9.115042	9.578951	9.589997	9.578951	16.96836	9.578951	9.126088											
ISURU-S/T-STND-SL-P3	8.559270726	8.6036760983	6.627597	6.594063	6.605164	6.605164	6.605164	14.16518	6.605164	6.57186											
ISURU-S/T-FR-P4	9.9512690561	10.1024040026	7.070866	1.386794	1.949666	1.397831	1.949666	142.4544	5.161344	1.386794											
ISURU-S/T-BA-P4	3.0054783715	12.49237105	7.022158	0.701029	7.032978	0.701029	0.701029	159.109	7.021947	0.71206											
ISURU-S/T-SL-P4	9.0036846964	5.4665718651	6.309377	3.936436	0.92984	3.947421	3.947421	13.48225	7.176959	7.176959											
ISURU-S/T-STND-FR-P4	1.5929195711	2.1062967665	11.06323	8.670971	8.670971	8.692817	8.681894	15.56333	8.670971	10.32033											
ISURU-S/T-STND-BA-P4	0.7749754588	0.1635227379	7.370155	0.076172	0.076172	0.108929	0.065254	14.74012	9.946766	0.054335											
ISURU-S/T-STND-SL-P4	0.8149556378	0.1016345235	8.023834	0.112546	0.112546	0.123458	0.123458	15.93464	10.67516	0.101635											
Sorted mean absolute errors table																					
Algorithm	Mean absolute error																				

Figure 3.15: Extract from a summary report generated automatically using the report generation framework

Algorithm 3.1 Pseudo-code of report generation framework

Input:
Path to folder containing data files
Path to folder containing reference data

```
for All files in data files folder do
    [reference_values] ← load_reference_data()
    data ← load_data_file()
    segment_radar_data_into_time_windows()
    find_the_corresponding_windows_for_reference_signals()
    for All time windows do                          ▷ This step is parallel
        [algorithm_names,results] ← ANALYZE(data)
        Reference_rate ← get_reference_rate(reference_data)
    end for
end for
create_LaTeX_report()
create_summary_excel_sheet()
function ANALYZE(data)
    Populate [algorithm_names] vector
    Populate [results] vector
    return [algorithm_names,results]
end function
```

3.6 Python and Matlab based real-time implementation

The data collection application described in 3.3 lacked the ability to process the data while it is being collected and for a real-world application, such as in a nursing home or an elder care facility where the system will be deployed, real-time processing of radar-data is critical as events related to respiration should be monitored as they happen. The real-time implementation in Matlab® which was developed as a potential solution to this was not as flexible as a software developed using a conventional programming language like C++ or Python due to the architecture of Matlab® as Matlab® is primarily optimized for data processing tasks instead of developing deploy-able applications. The architecture of the Parallel computing toolbox made it cumbersome to quickly launch and destroy different process and the lack of a threading framework (Parallel computing toolbox works by creating multiple processes instead of threads which are less resource intensive than separate processes[6] [39]) similar to that is available in languages such as C++, Python, Java, etc. makes it resource intensive to pass data between different processes

[6] https://www.mathworks.com/help/distcomp/

frequently as the different processes do not share a common memory space [39]. As a real-time application fetching data from a radar continuously and processing this data without slowing down the data acquisition would need at least two different threads/processes to collect data and process them, the amount of data that needs to be passed between different threads/processes is high.

Even though Matlab® offers a toolbox called Matlab Coder[7] to convert Matlab® code into C/C++ code, this toolbox does not support many functions used in the final system that would need to estimate breathing, show it on a user interface, send the data to the cloud to be shown on a web GUI, etc. and even if a system was somehow developed with only supported functions, the production cycle of converting each iteration into C/C++ and testing it on a deployment system adds a lot of complexity to a production environment. This is even more prominent in an embedded system as the final system was targeted to run in a Nvidia® Jetson™ TX2[8] single board computer running on an ARM® processor.A major consideration in such a system is the ability to be deployed easily and the cost effectiveness of the final system and easy deploy-ability of a Python application compared to a Matlab™ application and the free and open-source nature of many Python libraries means a Python based application is better suited for this task.

As seen on Figure 3.16, the real-time Python application starts by creating a Main process that is responsible for creating the other sub-processes/threads responsible for tasks such as collecting radar data, extracting breathing signal and estimating breathing rate, sending/showing results to the outside etc. The data collection thread connects to the radar (This thread is compatible with both Xethru® X4 radar and the FlatEarth® Ancho radar modules and the desired module can be selected by changing a configuration variable before running the application), this thread also has the capability to simulate a radar using pre-recorded data so that the application can be run to test out different algorithms/configurations on the exact same data. When in simulation mode, the collection thread reads a radar data csv file and plays back the data using the timestamps on the file which mimics a real radar exactly. The data is passed to the main thread through a Python thread safe queue data structure and after collecting enough data to estimate a breathing rate, the main process sends the data to a Matlab® process (using the Matlab® Engine API[9])

[7] https://www.mathworks.com/products/matlab-coder.html
[8] http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html
[9] https://www.mathworks.com/help/matlab/matlab-engine-for-python.html

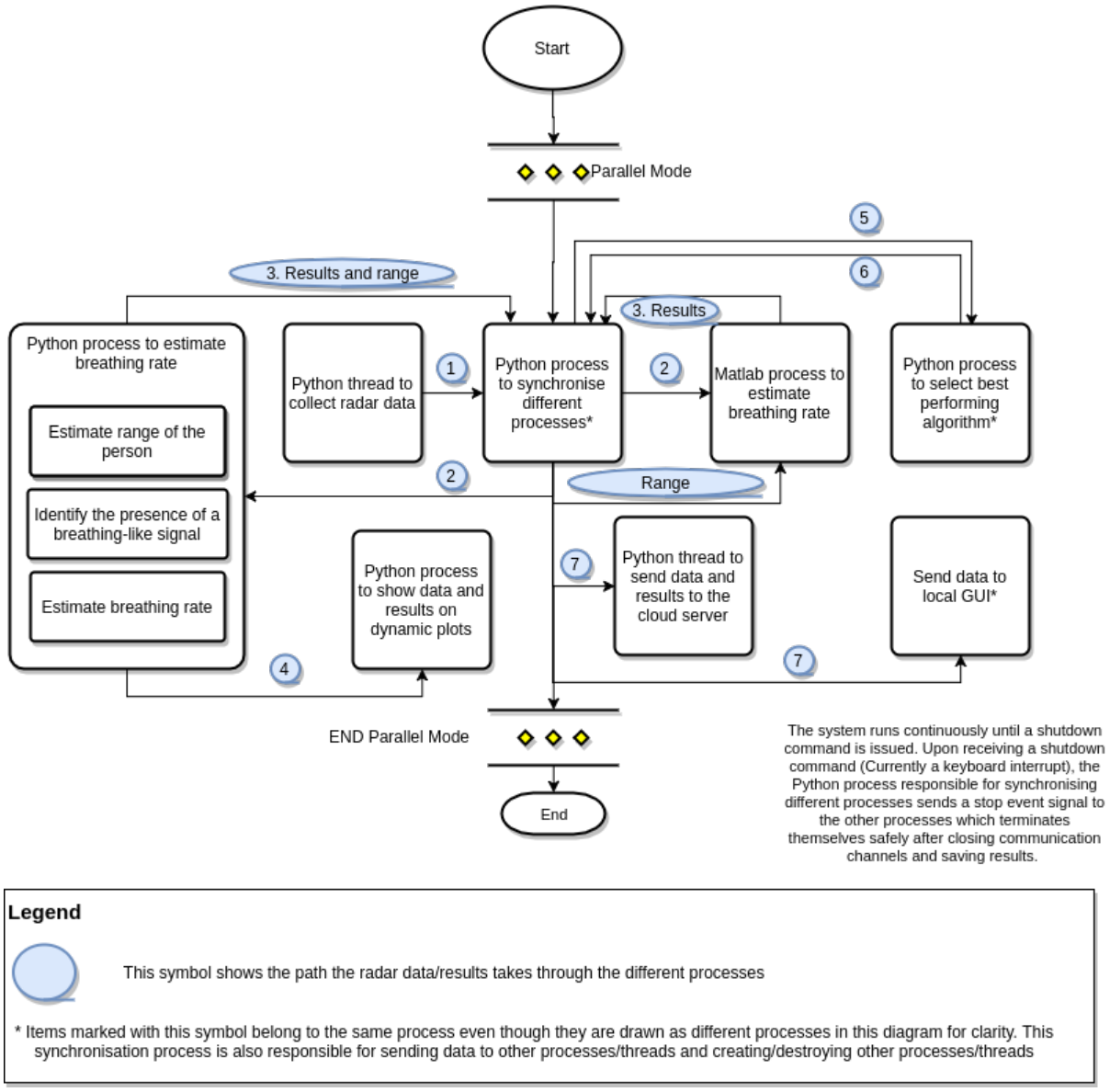


Figure 3.16: Organization of different processes in the real-time processing system

and a Python™ process simultaneously to estimate breathing. The reason for using a Matlab process here is for the system to be compatible with algorithms that were developed in Matlab®, mainly the NMD algorithm described in Section 3.2.3 is developed using an open source library in Matlab® provided by Iatsenko et al. [38]. The system is able to automatically identify if Matlab® is available in the hardware it is running on and automatically disables the Matlab® based algorithms if it detects that Matlab is not available. This makes it possible to run the system on multiple platforms/configurations without changing any of it's code or configuration. All the other features including the scoring algorithm presented in Section3.2.4 adapts to the presence or absence of Matlab automatically as well. As seen on Section 4.3, the EMD based

algorithm presented in Section 3.2.2 performs better than the NMD based algorithm and since all the algorithms used in the final system are developed using Python except the NMD algorithm, the presence or absence of Matlab® does not affect the system.

Breathing rates are calculated using a moving window algorithm where a window of 20 seconds of radar data is used with the time between each window being 2 seconds. The time between each window adjusts according to the load on the system and this enables the system to be run on even a low power embedded system such as a Raspberry Pi (The system was implemented on a Raspberry Pi 3B single board computer (with a limited number of algorithms for breathing rate estimation) with the time between each window adjusted to 3.5 seconds automatically.). Even though the time between each window is automatically adjusted, the window length is always a user defined constant which is not adjusted during run-time. The sampling rate of the radar, both in slow time and fast time are always constant and are not adjusted in any manner to maintain consistency of the results. After the data is processed and breathing signals and breathing rates are found, the results are sent to a separate thread that is responsible for showing the results on graphs so that a researcher can observe the signals and get an idea about the performance of the algorithms and possible improvements. The plotting functions are moved to a different thread so that the data collection and processing processes/threads do not get affected as plotting functions take a considerable amount of time if a lot of plots/information have to be shown. This also enables the plotting function to be disabled easily (Just one configuration variable needs to be adjusted to disable plotting) when the system is deployed as a final system. The plotter thread is also able to save all the plots on to the file system so that they can be viewed later on to see how the system performed.

After the main process gets all the data from Python and Matlab®(if enabled), it analyses all the results using the scoring system described in Section 3.2.4 and selects the best performing algorithm and reports that data onto a local GUI and also sends the data to a cloud server as a HTTP request. The inter-process and inter-thread communication between different Python threads/processes is done using thread safe Queues and the data is sent to a GUI using a network socket connection giving the flexibility to run the GUI on any computer in the local network (eg. at a nurses' station in a nursing home).

While doing all of the above tasks, the system also saves the raw radar data onto the local file

system as separate files once every minute (adjustable) so that the same data can be analyzed later on.

Figure 3.17 shows the CPU utilization on a system with 12 physical cores and 24 threads (Dual Intel® Xeon E5-2620 processors with 128GB RAM). As it can be seen on the top sub figure in Figure 3.17, all of the cores are utilized during operation and the periodic spikes in cpu usage corresponds to when the radar data is analyzed on a new set of data. Since radar data is processed only when enough data are collected to get a new breathing estimate, there are minimas in the cpu usage graph where the system is just collecting radar data or sending the previous results to the GUI or cloud. The system collects data for at least 1 second (adjustable) before trying to estimate a new breathing rate using the last 20 seconds (adjustable) of data. This creates a real-time moving window on the radar data which enables breathing rate to be estimated frequently even though human breathing signals are relatively slow (a usual breathing inspiration and respiration takes around 3-4 seconds for a healthy human). The system is also able to adapt itself automatically if it is run on low power hardware such as an embedded single board computer as the main process tracks the input and output queue sizes of the different processes such as the plotting thread, breathing estimation process, etc. and adjusts the frequency of estimation if it detects that the system is able to estimate breathing rate at the desired frequency of estimation. This adaptive system has been put into test when it was successfully run on a Nvidia® Jetson TX2 system[10]

Due to the data collection threads' ability to play-back csv files containing radar data mimicking the presence of a real-radar as described above, a special mode was added into this application to generate reports about the performance of different algorithms on a large set of data files. When running in this mode, the application loads the radar data files (As many files of varying lengths(time duration), sampling rates, etc. as desired) one by one along with reference data files (Captured using the data collection application described in Section 3.3) saved during the experiments and analyses the performance of algorithms for each test-case separately and saves the results in separate files corresponding to the multiple test-cases tested. This system allowed the researchers to do data collection during the day time and generate detailed reports

[10]The configuration and setup of the Nvidia Jetson Tx2 board was not done by the author but was done by two undergraduate students (Taylor Leblond and Qi Ye) at University of Ottawa after receiving initial instructions and the deploy-able code from the author.

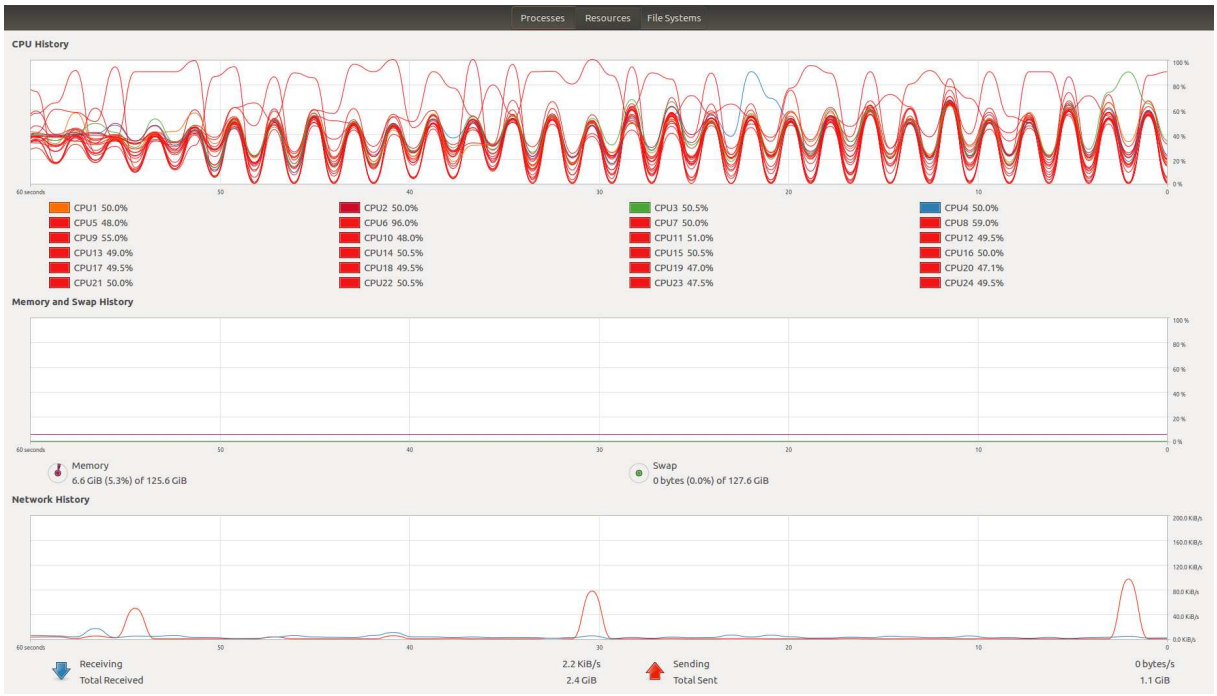


Figure 3.17: CPU usage when running the real-time Python application using multiple processes and threads.

(With pdf files containing graphs of results and csv files containing all the estimates, their mean error, variance, standard deviation, activity levels, etc.) overnight (or any time) by just running the system with the folder containing the radar data and the reference data files as input. An example extract from a pdf file created using this software can be found in Figure 3.18 and an extract from a summary report created using this software can be seen in Figure 3.19.

In addition to all of the above, the real-time application also includes the ability to start a Matlab® script containing the human activity and posture classification work by Zachary Baird described in [40] and get the results from it. This script is called in a similar method to how the breathing rate estimation algorithms in Matlab® are called as described above using the Matlab® Engine API[11] for Python™

[11] https://www.mathworks.com/help/matlab/matlab-engine-for-python.html

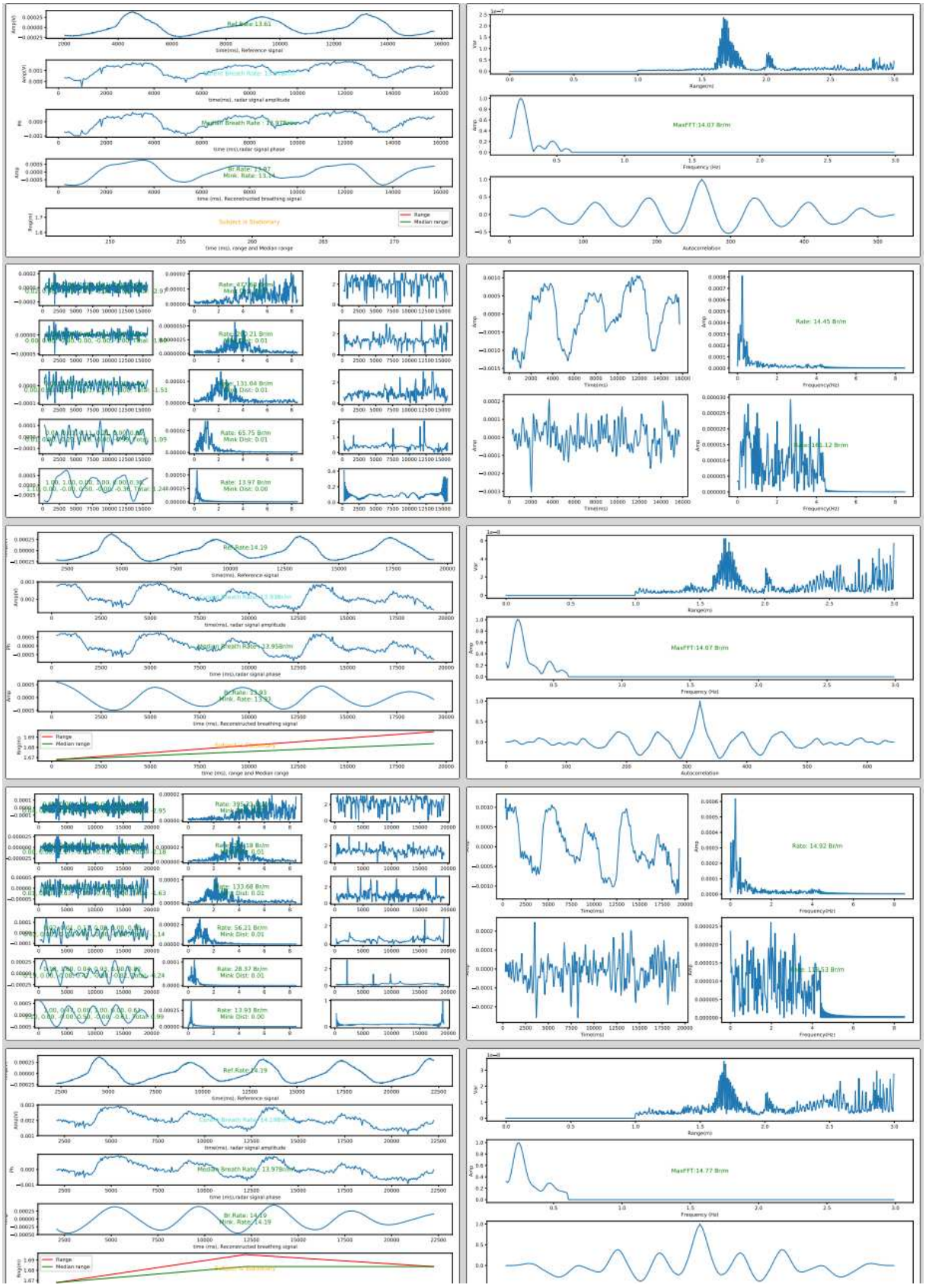


Figure 3.18: An extract from a pdf report generated using the real-time application

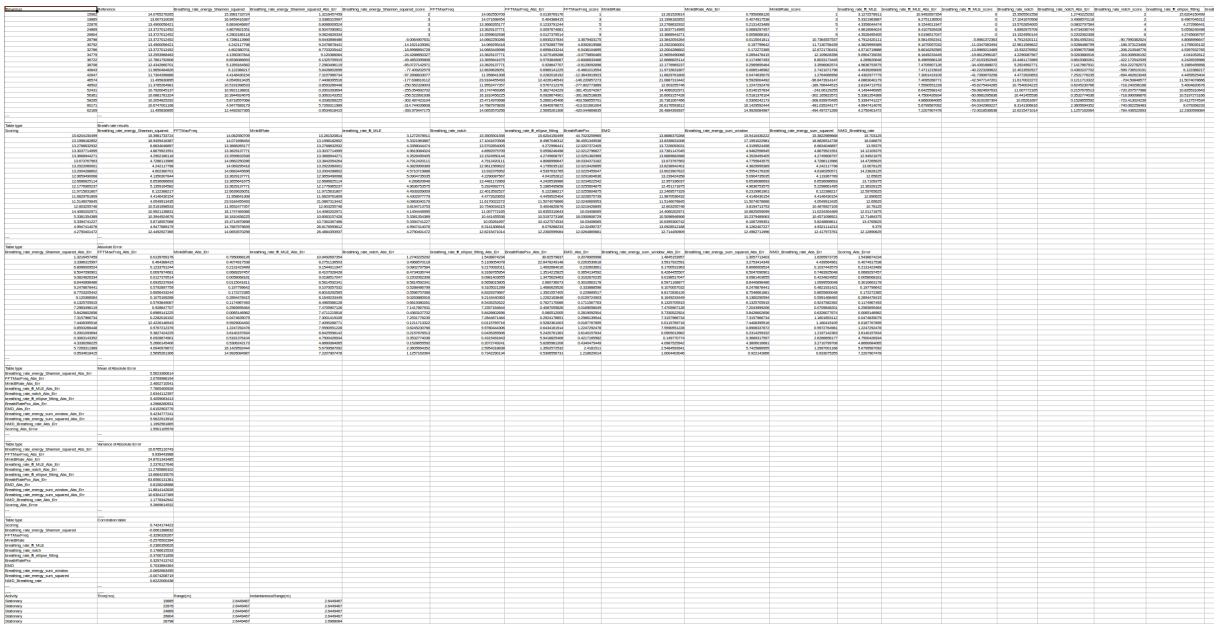


Figure 3.19: An extract from a csv summary report generated using the real-time application

Chapter 4

Results

This section analyses the results of the algorithms described in Chapters 2 and 3 with comparisons between instances when different algorithms perform well versus them performing adversely. The first part of this chapter (Section 4.2) focuses on range estimation and activity level detection described in Section 3.1 and the rest of this chapter (Section 4.3) focuses on results of breathing rate estimation using various algorithms.

4.1 Abbreviations used in this chapter

The following abbreviations/notations are used in this chapter to reduce clutter in graphs and tables:

- Reference: Reference signal obtained from a breathing belt
- EMD method: Novel CEEMDAN based method described in Section 3.2.2
- Minkowski distance method, 'MinkBRate': EMD and Minkowski distance based breathing rate estimation method proposed by Xinyang Zhang in [31]. 'MinkBRate' refers to the breathing rate calculated using this method.
- NMD: NMD based method for breathing rate estimation as described in Section 3.2.3
- FFT Max.Freq. method: FFT based breathing rate estimation method described in Section 3.2.1
- Scoring method: Automatic scoring method described in Section 3.2.4

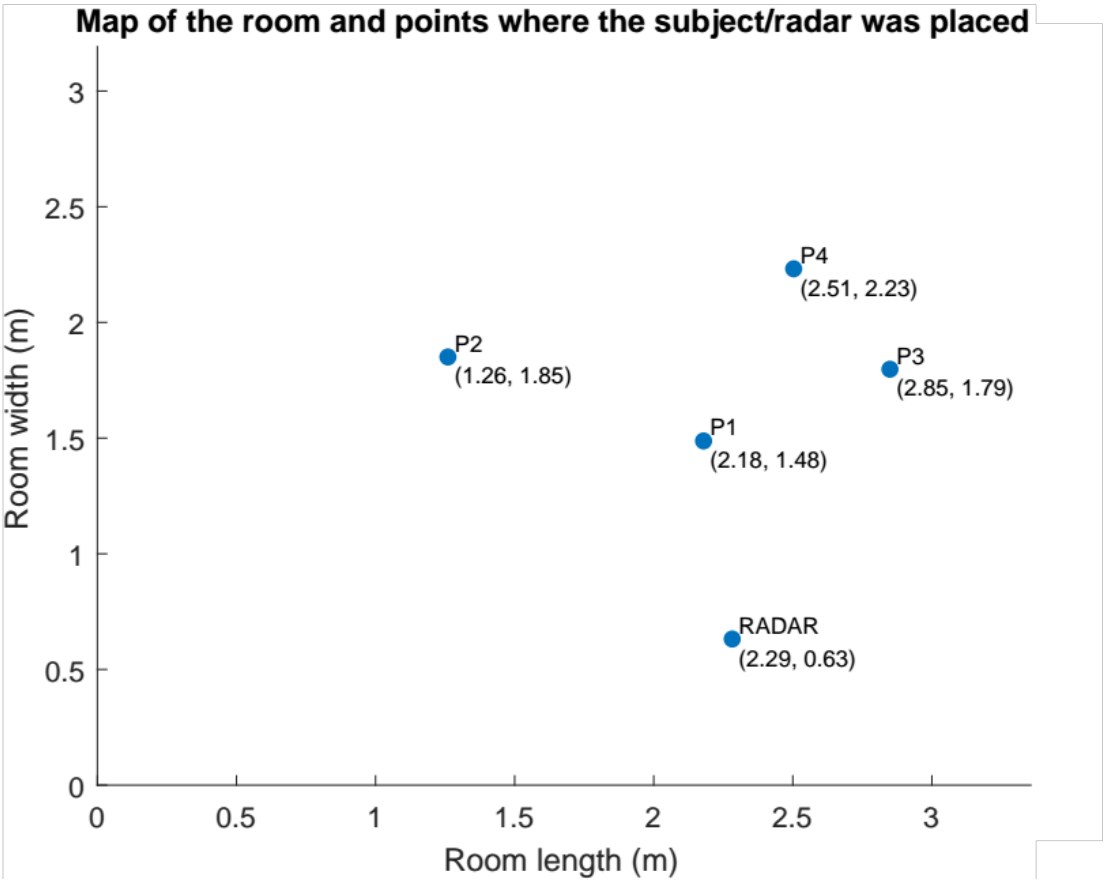


Figure 4.1: Positions of subjects and the radar

4.2 Range estimation

4.2.1 Stationary subject

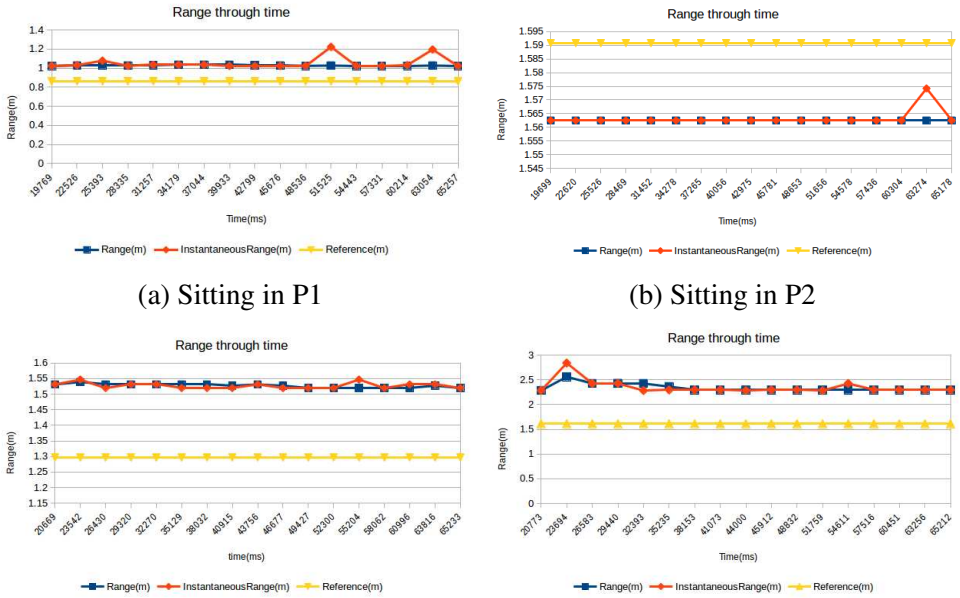
Range estimation of a stationary subject is explored through estimating the range of a subject placed in four locations inside a room with the subject facing their front, back and left side towards the radar with the subject standing or sitting down on a chair. A total of 24 test cases were observed for this (2 different postures and 3 different orientations (standing and front towards radar, sitting and front towards radar, standing and back towards radar, sitting and back towards radar, standing and side towards radar, sitting and side towards radar) in four different locations). The corresponding positions can be seen in Figure 4.1. The experiment was divided up into segments of 1 minute each with the subject changing posture/location once every minute and the automatic data collection system explained in Section 3.3 was used to automatically label and save the data in different files.

As seen on Figure 4.2, the subject location can be accurately tracked using the novel method described in Section 3.1. When the system detects that the subject is stationary, the median range of the n(adjustable) most recent instantaneous ranges are used as the range of the person to extract the breathing signal. A small offset is present between the reference range and estimated range as the reference ranges were taken by placing markers on the floor manually and the subjects were asked to sit/stand as close to the markers as possible and in most cases the subjects' chest is not positioned directly on top of the markers (eg: A sitting subject would have their legs on the markers with their chest further back from the marker).

Compared to the results from the range tracking method described in section 3.1, estimating range without tracking the person and calculating only once for the whole period of a test-case would give adverse results in some cases such as, 1.02m, 0.14m, 0.17m and 0.14m for range estimation for the person sitting at P1, P2, P3 and P4 with reference distances of 0.86m,1.59m,1.29m, and 1.61m respectively. This is due to the fact that when the person is selected based on maximum variance through time, noise from other events (such as noise from water pipes, vibrating radiators, etc.) show larger variances through time than human breathing for longer periods of time and the breathing signal of the person becomes insignificant compared to the accumulated different time varying signals and SVD based clutter removal or variance based range estimation can not properly differentiate between time varying noise and breathing signals.

Figure 4.2 and subsequent similar figures contain 4 sub-figures each corresponding to the four different positions P1,P2,P3 & P4 (as shown in Figure 4.1) and each plot contains 3 series' of data. Reference range is obtained manually by measuring while the instantaneous range is the range calculated in each time instance and the series named "range" is the time tracked range by taking the median of most recent instantaneous ranges as described in Section 3.1. The time tracked range value is used for extracting the breathing signal from the radar signal.

As seen on Figures 4.2, 4.3, 4.4, 4.5, 4.6, and 4.7, the radar is able to locate a stationary human very effectively, even in instances when the instantaneous range jumps due to noise or random body movements, the time tracked range, which is used to extract the breathing signal is not affected.



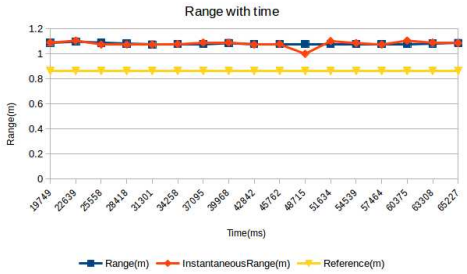
(a) Sitting in P1

(b) Sitting in P2

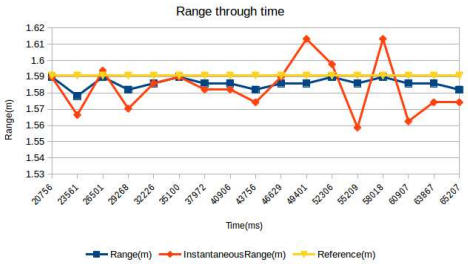
(c) Sitting in P3

(d) Sitting in P4

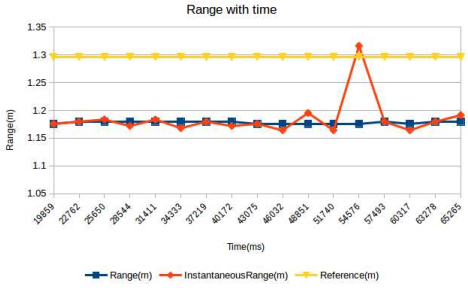
Figure 4.2: Range of subject with subject sitting in front of radar, with face towards radar



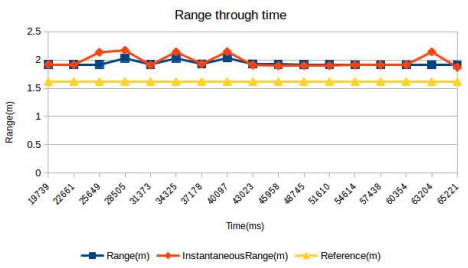
(a) Standing in P1



(b) Standing in P2



(c) Standing in P3



(d) Standing in P4

Figure 4.3: Range of subject with subject standing in front of radar, with front towards radar

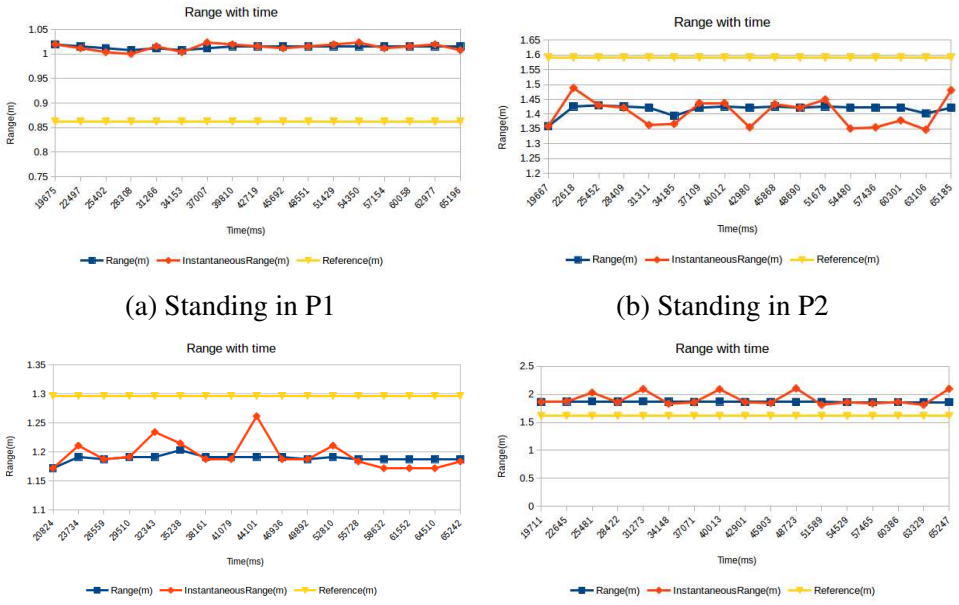
(a) Sitting in P1

(b) Sitting in P2

(c) Sitting in P3

(d) Sitting in P4

Figure 4.4: Range of subject with subject sitting in front of radar, with back towards radar



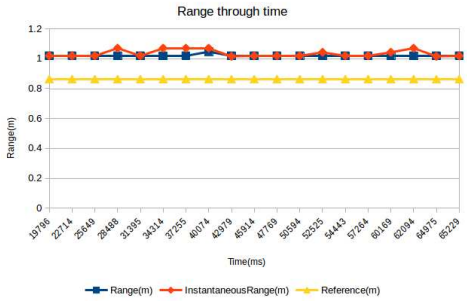
(a) Standing in P1

(b) Standing in P2

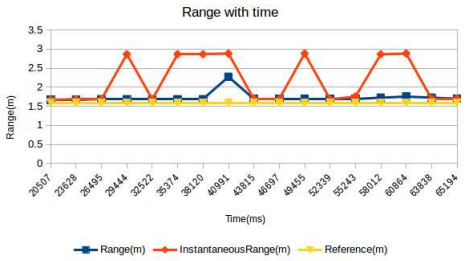
(c) Standing in P3

(d) Standing in P4

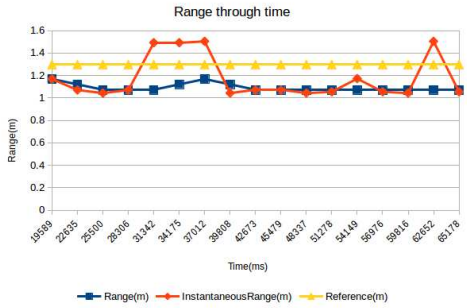
Figure 4.5: Range of subject with subject standing in front of radar, with back towards radar



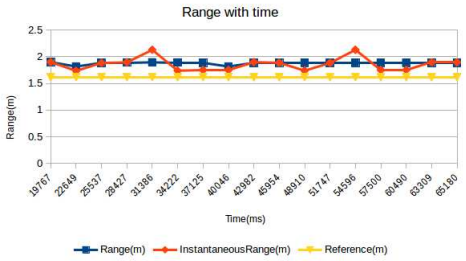
(a) Sitting in P1



(b) Sitting in P2

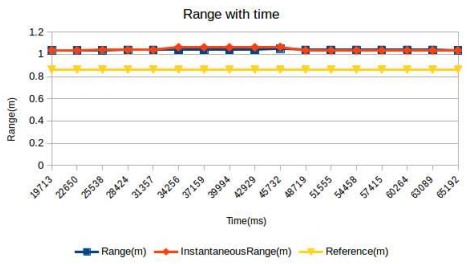


(c) Sitting in P3

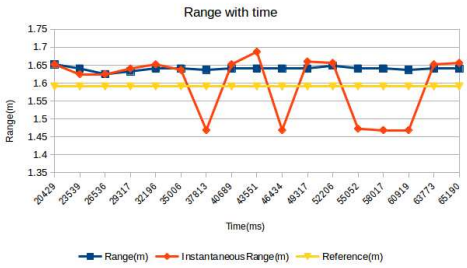


(d) Sitting in P4

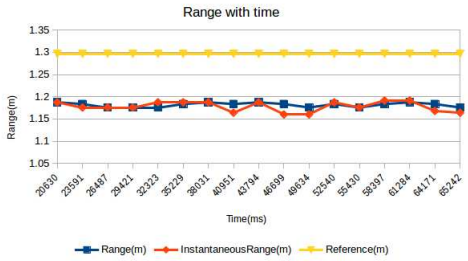
Figure 4.6: Range of subject with subject sitting in front of radar, with left side towards radar



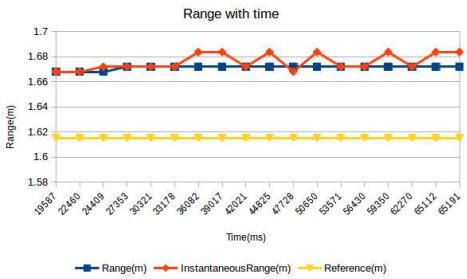
(a) Standing in P1



(b) Standing in P2



(c) Standing in P3



(d) Standing in P4

Figure 4.7: Range of subject with subject standing in front of radar, with left side towards radar

4.2.1.1 Summary of results

Since the subject is stationary in all test cases (Given that the subject behaves as instructed and stays sufficiently stationary), the expected value of the standard deviation of the range estimation should be 0 for all test cases. Tables 4.1, 4.2 and 4.3 shows the standard deviation of the tracked range, (tracked using the method described in Section 3.1) and the instantaneous range, calculated for all test cases. It can be seen that the standard deviation values are very close to 0 for all test cases except when the subject is sitting at P2 with their back or left side towards the radar. This is due to some outliers in range estimation as it can be seen in Figures 4.4 and 4.6. However, it can be seen that the tracked range estimation minimizes the standard deviation in these two cases as well allowing the system to track the subject as effectively as possible.

The range estimation is also evaluated by comparing it to the reference range value obtained using a distance measurement device. Since a bias value is present in all range estimations from the radar and the manually collected distance to the persons location due to the reasons described in the start of Section 4.2.1, the summary of the results contain the mean absolute errors of the instantaneous range and tracked range before and after removing the bias value. The bias value is removed by taking the mean error rate and removing this mean value from the estimated range. Tables 4.4, 4.5, 4.6, 4.7, 4.8, and 4.9 contains the summary of range estimation of a stationary subject and as seen from these tables, the system is able to track the subject properly in all test-cases. However, it should be noted here that the same two test cases mentioned above (Sitting with back towards radar on position P2 and sitting with left side towards radar on position P2) shows higher mean error in both tracked and instantaneous ranges.

Table 4.1: Standard deviation of range estimation for all test cases with subject facing the radar

	P1	P2	P3	P4
Sitting and facing the radar				
Standard deviation of tracked range (m)	0.00	0.00	0.01	0.08
Standard deviation of instantaneous range (m)	0.06	0.00	0.01	0.14
Standing and facing radar				
Standard deviation of tracked range (m)	0.01	0.00	0.00	0.04
Standard deviation of instantaneous range (m)	0.02	0.02	0.04	0.11

Table 4.2: Standard deviation of range estimation for all test cases with subjects' back towards the radar

	P1	P2	P3	P4
Sitting with back towards radar				
Standard deviation of tracked range (m)	0.00	0.59	0.00	0.00
Standard deviation of instantaneous range (m)	0.09	0.79	0.45	0.01
Standing with back towards radar				
Standard deviation of tracked range (m)	0.00	0.02	0.01	0.01
Standard deviation of instantaneous range (m)	0.01	0.05	0.02	0.11

Table 4.3: Standard deviation of range estimation for all test cases with subjects' left side towards the radar

	P1	P2	P3	P4
Sitting and side towards the radar				
Standard deviation of tracked range (m)	0.01	0.14	0.04	0.02
Standard deviation of instantaneous range (m)	0.02	0.60	0.19	0.12
Standing and side towards the radar				
Standard deviation of tracked range (m)	0.00	0.01	0.00	0.00
Standard deviation of instantaneous range (m)	0.01	0.09	0.01	0.01

Table 4.4: Error rates in range estimation for all experiments with person sitting in front and facing the radar

Position	Mean error in tracked range (m)	Mean error in instantaneous range (m)	Bias removed mean error in tracked range (m)	Bias removed mean erorr in instantaneous range (m)
P1	0.17	0.19	0.004	0.04
P2	0.03	0.03	0.00	0.00
P3	0.23	0.23	0.005	0.01
P4	0.72	0.73	0.06	0.09

Table 4.5: Error rates in range estimation for all experiments with person standing in front and facing the radar

Position	Mean error in tracked range (m)	Mean error in instantaneous range (m)	Bias removed mean error in tracked range (m)	Bias removed mean erorr in instantaneous range (m)
P1	0.22	0.22	0.005	0.01
P2	0.00	0.01	0.002	0.01
P3	0.12	0.11	0.002	0.01
P4	0.32	0.36	0.03	0.10

Table 4.6: Error rates in range estimation for all experiments with person sitting with back towards the radar

Position	Mean error in tracked range (m)	Mean error in instantaneous range (m)	Bias removed mean error in tracked range (m)	Bias removed mean erorr in instantaneous range (m)
P1	0.15	0.19	0.000	0.05
P2	0.85	0.79	0.42	0.48
P3	0.28	0.35	0.002	0.14
P4	0.02	0.03	0.00	0.01

Table 4.7: Error rates in range estimation for all experiments with person standing with back towards the radar

Position	Mean error in tracked range (m)	Mean error in instantaneous range (m)	Bias removed mean error in tracked range (m)	Bias removed mean erorr in instantaneous range (m)
P1	0.15	0.15	0.002	0.01
P2	0.17	0.19	0.011	0.04
P3	0.11	0.10	0.004	0.02
P4	0.25	0.30	0.00	0.10

Table 4.8: Error rates in range estimation for all experiments with person sitting with their left side towards the radar

Position	Mean error in tracked range (m)	Mean error in instantaneous range (m)	Bias removed mean error in tracked range (m)	Bias removed mean erorr in instantaneous range (m)
P1	0.16	0.17	0.003	0.02
P2	0.14	0.59	0.07	0.57
P3	0.21	0.22	0.029	0.03
P4	0.26	0.24	0.01	0.09

Table 4.9: Error rates in range estimation for all experiments with person standing with their left side towards the radar

Position	Mean error in tracked range (m)	Mean error in instantaneous range (m)	Bias removed mean error in tracked range (m)	Bias removed mean erorr in instantaneous range (m)
P1	0.18	0.18	0.002	0.01
P2	0.05	0.08	0.003	0.03
P3	0.11	0.12	0.004	0.01
P4	0.06	0.06	0.00	0.01

4.2.2 Non-stationary subject

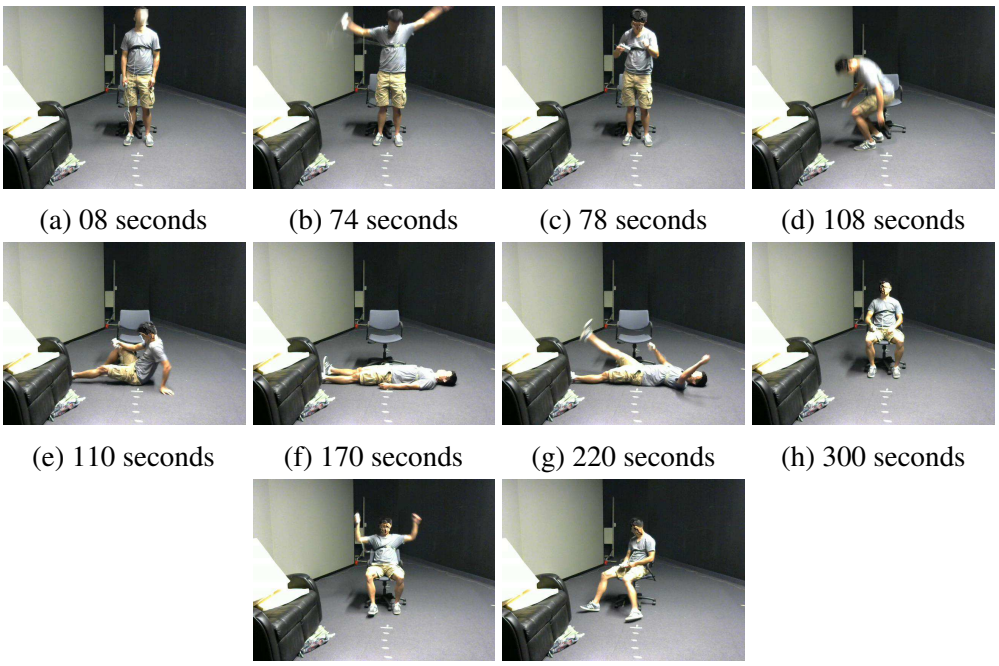
Range estimation of non stationary subjects are explored using two different scenarios:

1. Subject changing postures randomly such as sitting, standing, lying down, moving arms/legs while sitting, standing or lying down, etc.

2. Subject walking around the room.

4.2.2.1 Subject changing postures randomly

In this experiment the subject was asked to change their posture randomly and to move their arms/legs randomly from time to time. Some of the movements can be seen on Figure 4.8 (The camera is placed at 5 cm to the left side of the radar and the camera and radar are both oriented at the same direction). As seen on Figure 4.9, the system is able to locate the person always. The subject is standing at around 4.5 meters from the radar and as it can be seen on Figure 4.9, when the subject sits down (at around 250 seconds) on the chair, his range increases to 5 meters as his chest is farther back from the radar when he's sitting. It can also be seen on Figure 4.9, the subjects' level of activity is properly detected by the radar by the algorithm described in Section 3.1.2. It can also be seen that the activity level detection is accurate in 171 out of 178 total cases and this translates into an accuracy of 96%.



(a) 08 seconds (b) 74 seconds (c) 78 seconds (d) 108 seconds

(e) 110 seconds (f) 170 seconds (g) 220 seconds (h) 300 seconds

(i) 330 seconds (j) 350 seconds

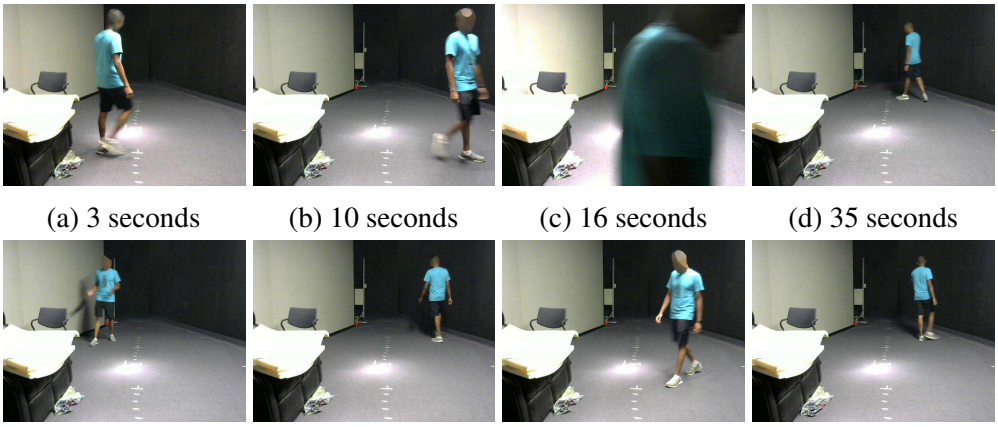
Figure 4.8: Some of the movements of the subject with time (Faces blurred for anonymity). The movements depicted in the Figures are: (a) Stationary and standing, (b) and (c) Standing with random body movements (RBM), (d) and (e) lying down with RBM, (f) lying down stationary (g) lying down with RBM, (h) sitting down stationary, (i) and (j) sitting with RBM

Figure 4.9: Range tracking of subject in different postures

4.2.2.2 Subject walking around the room

In this experiment the subject is walking back and forth from the radar randomly. The subject does not walk in a straight line and moves around the room randomly. However the subjects' movements are always back and forth towards and away from the radar along the radars' axis. (*i.e.* Let's assume the radar is facing north. The subject is them always moving towards north and south in front of the radar(however, the distance they move towards north and south are random). But his movements towards east and west are random) Hence, if the radar is tracking the subject properly, the range profile should show the range of the subject increasing and decreasing forming a zigzag pattern. The subject is also free to move their arms and talk while they are moving. As seen on Figures 4.10, 4.12 and 4.14, the subjects are completely free to walk around the room as they wish.

As seen on Figures 4.11, 4.13, and 4.15, the range tracking system described in Section 3.1 is able to track the range of the subject as a very prominent zigzag pattern is seen on the range profile, and it can also be seen that the movement detection algorithm described in Section 3.1.2 is able to detect that the person is not stationary 100% of the time as there are no instances in Figures 4.11, 4.13, and 4.15 when the person is detected as stationary. Even though the range tracking system described in Section 3.1 runs in the background, the tracked range is never used when the person is not stationary and the breathing rate is not calculated in such instances.



(a) 3 seconds (b) 10 seconds (c) 16 seconds (d) 35 seconds

(e) 52 seconds (f) 72 seconds (g) 139 seconds (h) 173 seconds

Figure 4.10: Some of the movements of subject 1 with time (The subject is moving back and forth towards and away from the radar continuously)

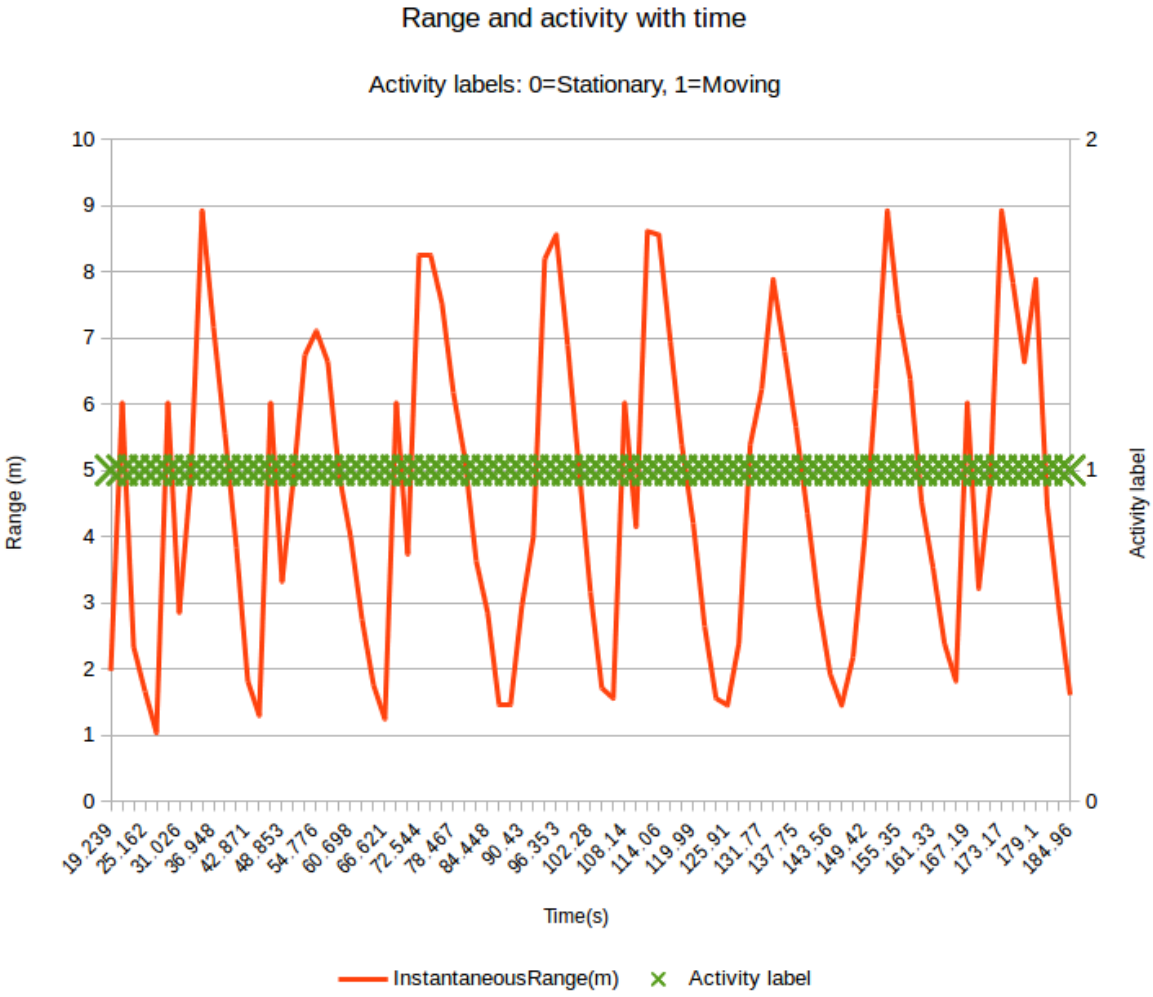
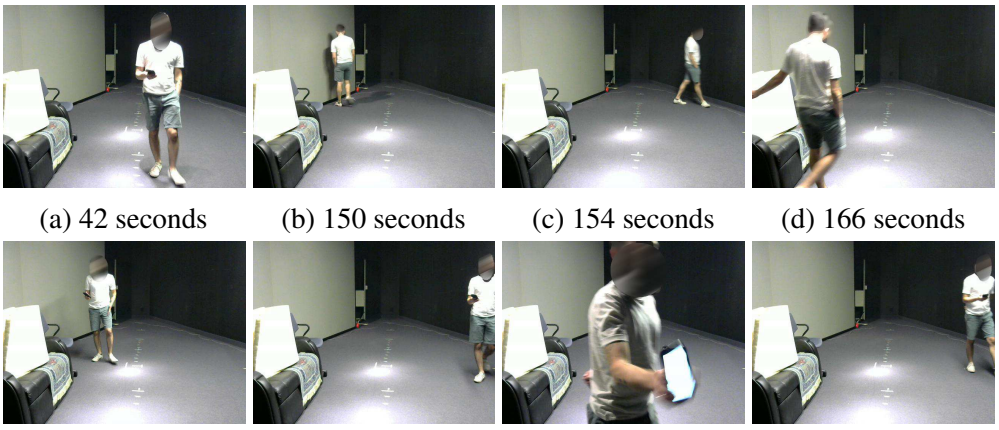


Figure 4.11: Range and activity tracking with subject 1 walking around the room



(a) 42 seconds (b) 150 seconds (c) 154 seconds (d) 166 seconds

(e) 286 seconds (f) 307 seconds (g) 314 seconds (h) 350 seconds

Figure 4.12: Some of the movements of subject 2 with time (The subject is moving back and forth towards and away from the radar continuously)

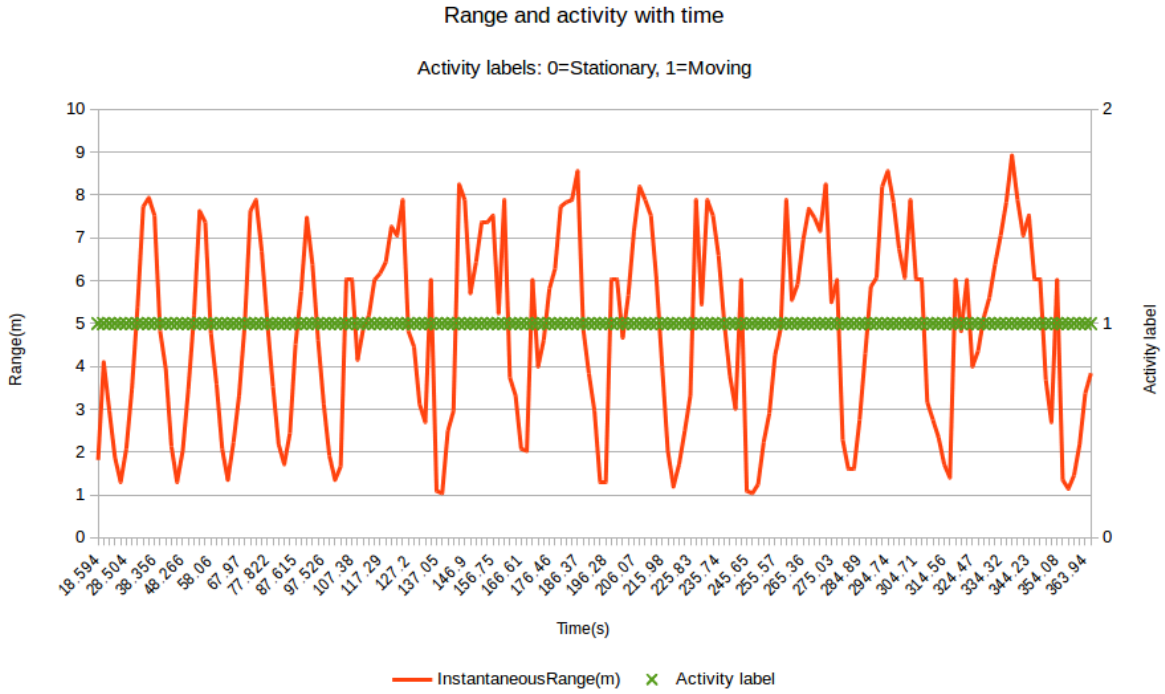


Figure 4.13: Range and activity tracking with subject 2 walking around the room

(a) 3 seconds (b) 81 seconds (c) 87 seconds (d) 155 seconds

Figure 4.14: Some of the movements of subject 3 with time (The subject is moving back and forth towards and away from the radar continuously)

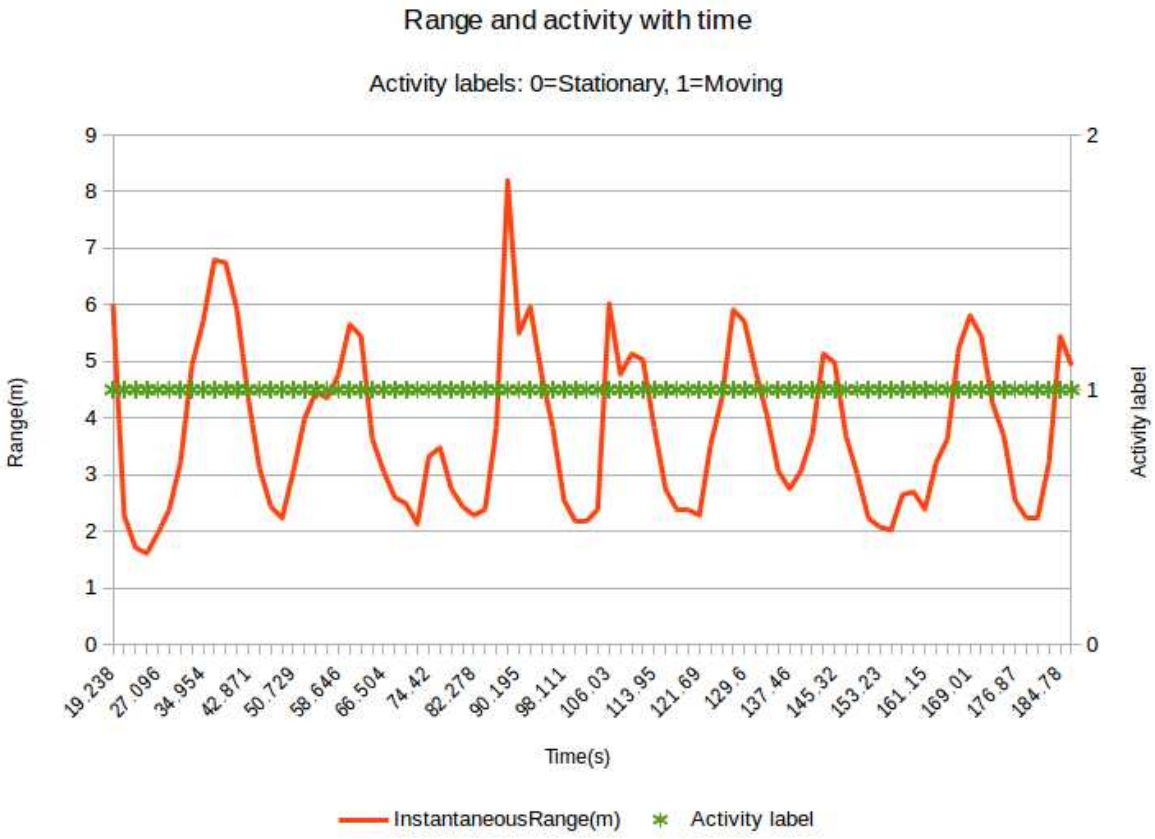


Figure 4.15: Range and activity tracking with subject 3 walking around the room

4.3 Breathing rate estimation

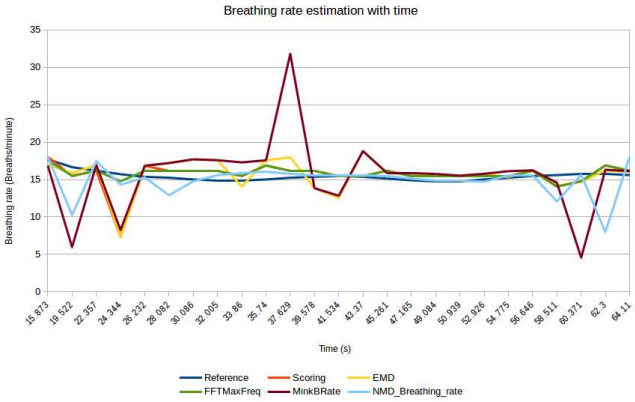
This section examines breathing rate estimation in a wide variety of test cases. The performance of the four best performing breathing rate algorithms, namely, the novel CEEMDAN based method (Referred to as EMD method in this section) introduced in Section 3.2.2, FFT based method described in Section 3.2.1, Minkowski distance based method introduced by Xinyang Zhang in [31] and described in Section 2.3.3 (Referred to as MinkBRate method in this section) and the NMD based method described in Section 3.2.3 and the results of the scoring algorithm described in Section 3.2.4 which selects the best algorithm dynamically are the main focus of this section. Some results from other algorithms are also shown in Section 4.3.2.

4.3.1 Reference value calculation

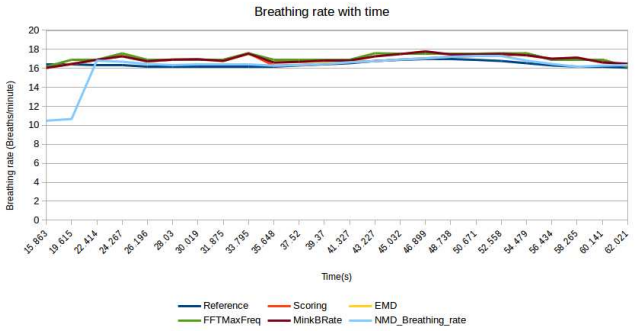
The reference values used in this section (Section 4.3) are calculated using a BioRadio™ 150 breathing belt sensor by GLneurotech®. A C++ application was developed using the C++ API provided by GLneurotech as described in Section 3.3 and the data from the breathing belt sensor is synchronized with the radar data by opening a socket connection between the Matlab application (Section 3.3) collecting data and the C++ application and transmitting relative time-stamps (Times are synchronized once per test-case and the two applications keep track of the relative time-stamps throughout each test-case). Since the data processing is done using overlapping windows, after selecting the window of data corresponding to the radar data, the closest possible window is found corresponding to the data from the breathing belt file. This has to be done since the two sensors have different sampling rates and the window start and end times are always slightly different (in the order of few milliseconds). After selecting this window of breathing belt data, the Fourier transform of the data is calculated after applying a Hamming window onto the data. The frequency value corresponding to the maximum amplitude in the frequency domain is selected and this frequency value is multiplied by 60 to estimate the breathing rate. Since the data from the breathing belt are very clean compared to the data from the radar, such a simple method can be used for estimating the breathing rate from the reference data. As mentioned on Section 3.6, the breathing rates are calculated using a moving window algorithm for all algorithms, including the reference rate, with the window size being 20 seconds.

4.3.2 Stationary subject at multiple positions and orientations

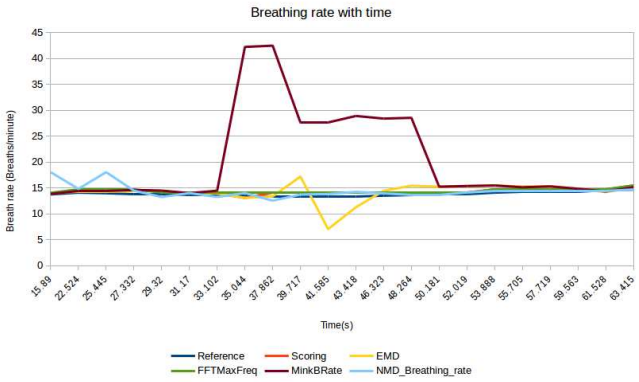
This section contains the breathing rate estimation results from the same set of test-cases as described in Section 4.2.1. Figure 4.16 shows the system tracking the breathing rate of the subject through time when the subject is sitting in front of the radar with their face towards the radar at the four points described in Section 4.2.1. The summary of all the test-cases can be seen in Tables B.1 to B.24. As seen on these tables, a single algorithm can not be pointed out to be working well in all the cases as the performance of algorithms vary between test cases. As seen on these cases, the scoring system is able to reduce the error rate of the final estimate in most of the cases except when all the algorithms perform poorly. This is due to the fact that the scoring system relies on the assumption that at least a majority of the used algorithms give a good enough estimate. However, when majority of algorithms give a good enough estimate, the scoring system is able to select the best algorithm and reduce the mean error rate as seen on Tables B.1 to B.24 in Appendix B.1. Table 4.10 shows a summary of all test cases with a stationary subject and as seen on this table, the mean error rate of the breathing rate estimation is relatively higher when the person is standing compared to the cases when the person is sitting down. This is due to the fact that the persons' body moves slightly during such instances and hence this affects the breathing rate estimation as the chest movements are modulated on top of the small body movements. Even though the novel CEEMDAN based algorithm described in Section 3.2.2 is able to decompose large body movements into different modes if the body movements have relatively larger amplitude compared to chest movements, it is not able to differentiate between body movements and chest movements if both the body and chest movements have similar amplitudes (If the body movements are too large obstructing the radar from seeing a breathing signal, the movement detection algorithm in Section 3.1.2 detects such movements and classifies these instances as the subject moving and breathing rates are not calculated). Therefore a possible improvement on the breathing rate estimation algorithms would be to detect when breathing signals are modulated with small body movements and remove such movements from the signal before estimating breathing rate. One possible solution would be to develop a system that learns a subjects' breathing patterns with time and is able to differentiate between a radar signals with and without modulation from small body movements.



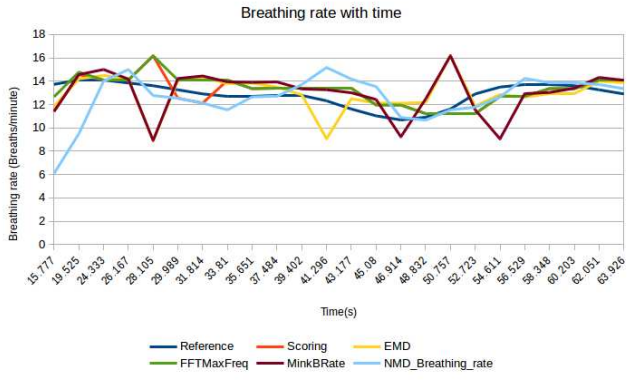
(a) Sitting in point 1



(b) Sitting in point 2



(c) Sitting in point 3



(d) Sitting in point 4

Figure 4.16: Breathing rate of subject with subject sitting in front of radar, with their front towards radar

Table 4.10: Standard deviation of range estimation (in meters) and mean error rate of breathing rate (using the scoring algorithm in Section 3.2.4) estimation (in breaths per minute) for all positions and orientations for stationary subject

Orientation/Posture	P1		P2		P3		P4	
	Range	B.Rate	Range	B.Rate	Range	B.Rate	Range	B.Rate
FRONT								
SIT	0	1.11	0	0.44	0.01	0.51	0.08	0.92
STAND	0.01	2.24	0	2.23	0	3.06	0.04	1.72
BACK								
SIT	0	0.64	0.59	1.68	0	3.99	0	2.04
STAND	0	3.28	0.02	1.91	0.01	6.11	0.01	0.68
SIDE								
SIT	0.01	1.99	0.14	0.8	0.04	0.45	0.02	4.39
STAND	0	2.68	0.01	1.76	0	4.13	0	0.86

4.3.2.1 Performance of other miscellaneous algorithms

This subsection contains the results from some of the algorithms described in Section 2.3. The algorithms compared in this section are:

- Shannon energy based algorithm in Section 2.3.1.
- Notch filter based algorithm in Section 2.3.1 [17].
- Ellipse fitting algorithm in Section 2.3.1.
- Wavelet transform based method in Section 2.3.3 [30].
- Welch periodogram based method in Section 2.3.1.
- Rectangular moving window algorithm in Section 2.3.1.
- Sum squared algorithm in Section 2.3.1.

The results corresponding to this section can be found in Appendix B.2. As seen in Tables B.25 to B.30, these algorithms work adequately only in some of the test cases and it can not be reliably said that any one of these algorithms work well in all the test cases. Hence, these algorithms are not included in the rest of this chapter for comparison of accuracy.

4.3.3 Comprehensive study with multiple subjects

A comprehensive study was conducted with 9 subjects sitting (with their front towards the radar) in front of the radar on a swivel chair approximately 4 meters away from the radar with a breathing belt attached to their chest and breathing normally. The goal of this experiment was to test the accuracy of the breathing rate estimation system for different subjects and to see if there are any noticeable differences in accuracy of the algorithms for the different subjects. The results of this experiment are presented in this section. At the time of the experiments, the subjects were males between 18 and 35 years of age and to the best of the authors' knowledge (based on voluntary disclosure by subjects) did not have any respiratory diseases. The results and the performance of algorithms in different conditions are explained using results from 3 subjects in this section and Section 4.3.3.4 contains a summary of the results for all 9 subjects.

4.3.3.1 Subject 01

As seen on Figure 4.17, the novel EMD based algorithm presented in section3.2.2 closely follows the reference breathing rate and it can be seen that the other methods such as the Minkowski distance based method used by Xingyang Zhang in [31], described in section2.3.3 makes significant errors in estimating breathing rate in some situations. However, the scoring method introduced in 3.2.4 is able to select the best performing algorithm at all times as it can be seen that the result from the scoring algorithm closely follows the reference breathing rate. This can be further observed in table 4.11 where it can clearly be seen that the scoring method has the lowest mean absolute error.

An assumption in the scoring algorithm to select the best performing algorithm automatically is that the breathing rate does not change suddenly. However, if the breathing rate changes

Table 4.11: Mean absolute error and standard deviation of breathing rate estimation error (units are in breaths per minute) of best performing algorithms

Algorithm	Mean absolute error	Standard deviation of absolute error
EMD method	2.11	3.93
FFT Max.Freq. method	2.33	3.89
Minkowski distance method	3.96	5.09
NMD method	1.98	2.81
Scoring method	1.66	2.66

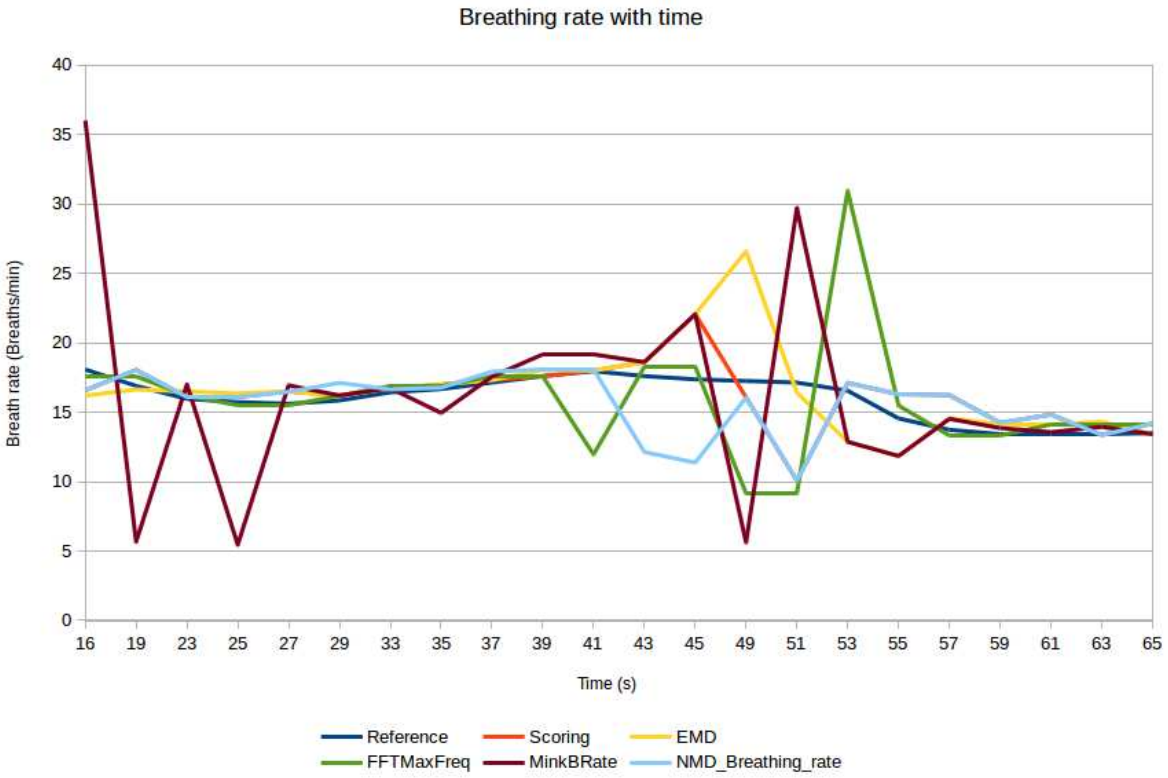


Figure 4.17: Breathing rate of subject 01 with time

gradually the scoring algorithm should be able to track and select the best performing algorithm. Figure 4.18 shows such a situation when the breathing rate changes with time and the scoring algorithm closely tracking the breathing rate.

Figure 4.19 shows a situation when the subject moves their body randomly(small head and arm movements). The subject moves their head and arms near the end of the experiment and it can be seen that their breathing rate also changes during this time. However, as seen on this figure, the breathing rate estimation is not severely affected by this as the result after the scoring method closely follows the reference breathing rate. Table 4.12 shows the mean absolute error and the standard deviation of the absolute error which reiterates that the breathing rate estimations are performing well in this test-case as well.

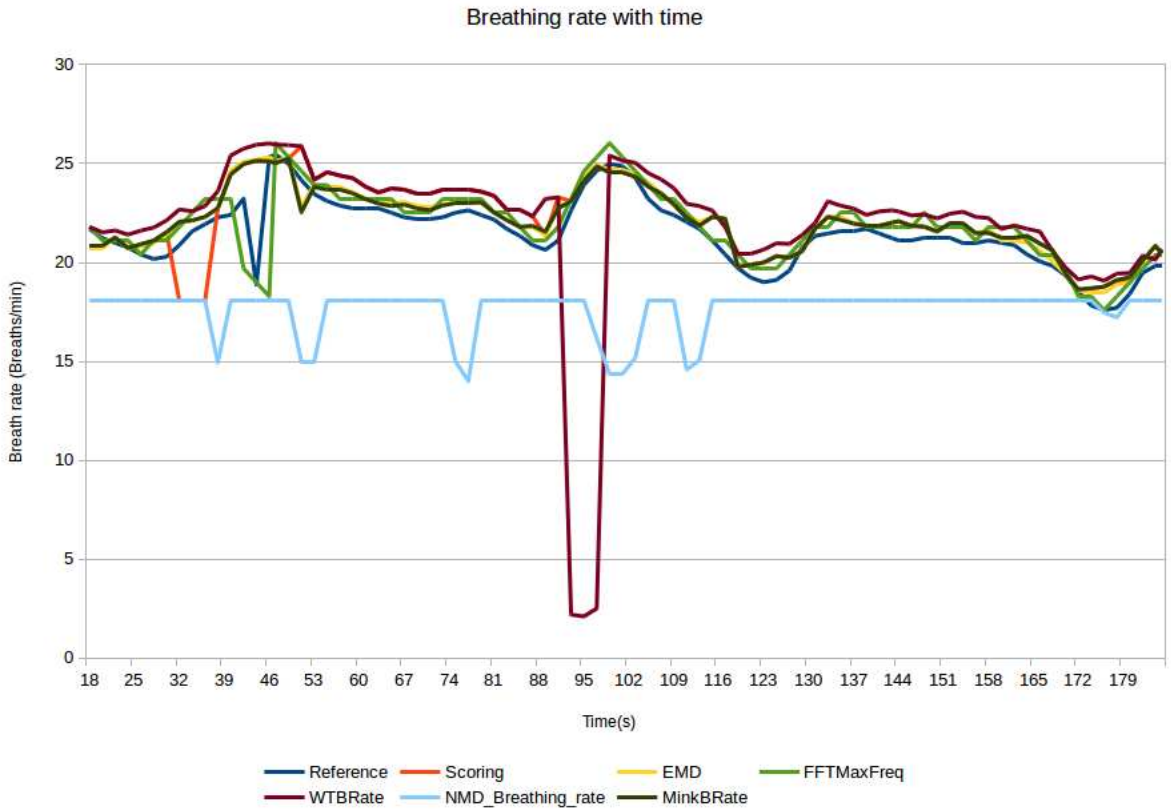


Figure 4.18: Effect of changing breathing rates on the scoring algorithm

Table 4.12: Mean absolute error and standard deviation of absolute error (Units are in breaths/minute) of breathing rate estimation for a test-case where the subject performs small random body movements (test-case shown in Figure 4.19)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	1.14	1.50
FFTMaxFreq method	1.23	1.71
MinkBRate method	3.00	4.62
NMD method	4.46	2.14
Scoring method	1.23	1.69

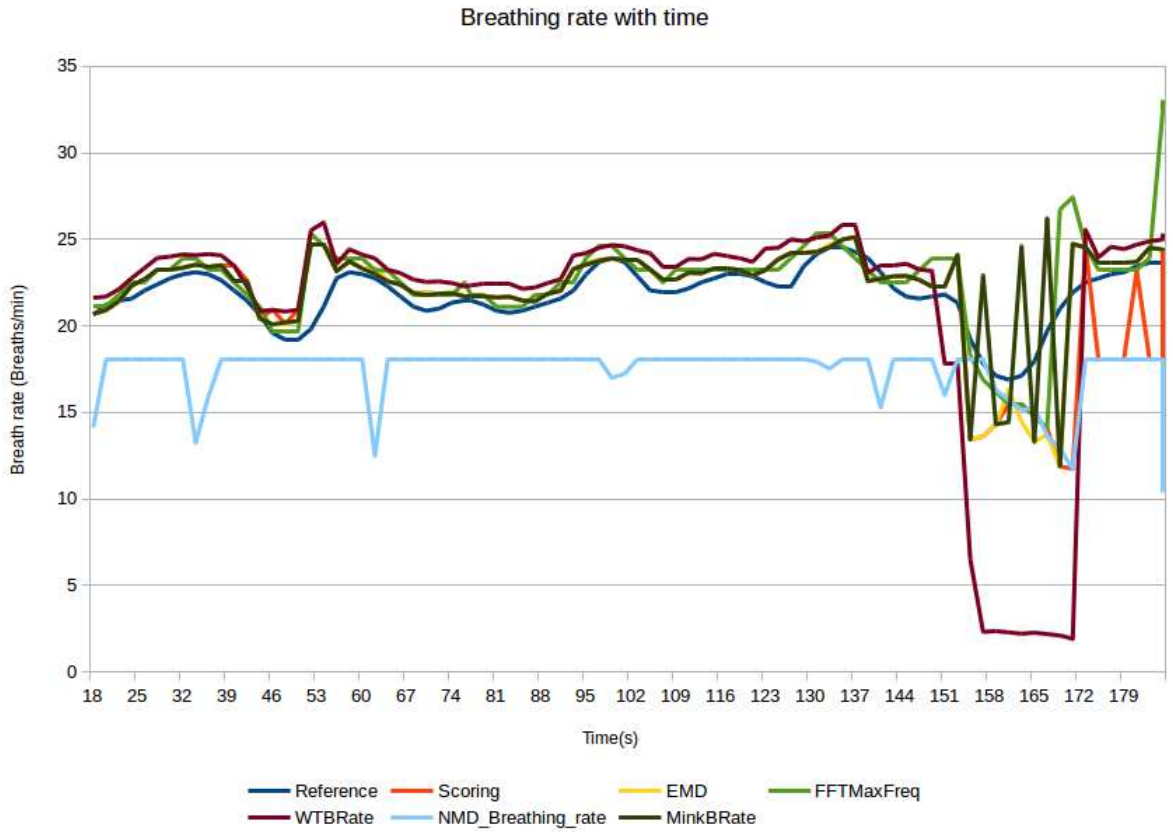


Figure 4.19: Effect of random body movements on breathing rate

4.3.3.2 Subject 02

As it can be seen from Figure 4.20, the subject's breathing rate is tracked properly by most of the algorithms except the Minkowski distance based method and the FFT based method during the period around 45 seconds to 52 seconds. As it can be seen on the figure, the scoring system is still able to select the correct breathing rate in this situation too. This is also apparent in the absolute error rate of the scoring method as shown in Table 4.13. Figure 4.21 shows a situation when the FFT based method continuously gives very high estimates for breathing rate. However,

Table 4.13: Mean and standard deviation of absolute error (in breaths per minute) of best performing algorithms for the test-case corresponding to Figure 4.20

Algorithm name	Mean Absolute Error	Standard deviation of Absolute Error
EMD method	0.49	0.60
FFT Max.Freq. Method	1.93	3.08
Minkowski distance method	1.62	3.75
NMD method	1.51	1.87
Scoring method	0.48	0.62

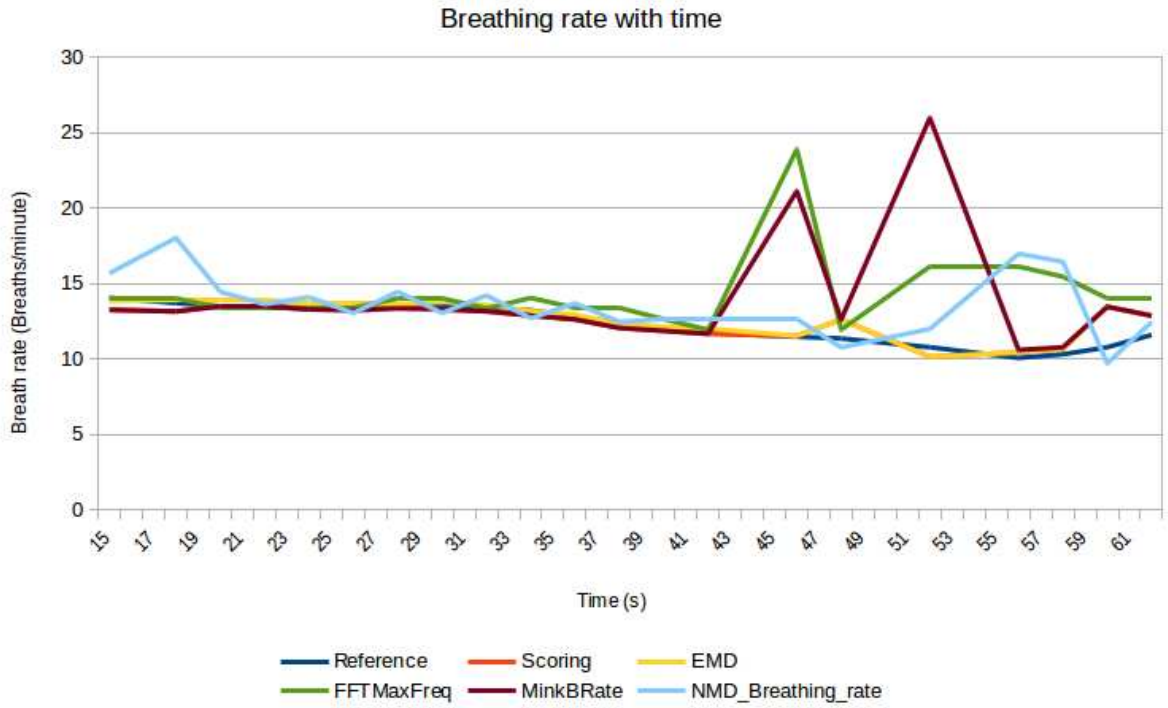


Figure 4.20: Breathing rate tracked through time

Table 4.14: Mean and standard deviation of absolute error (in breaths per minute) of best performing algorithms for the test-case corresponding to Figure 4.21

Algorithm name	Mean Absolute Error	Standard deviation of Absolute Error
EMD method	1.24	2.86
FFT Max.Freq. Method	7.42	8.36
Minkowski distance method	1.62	3.50
NMD method	2.06	2.78
Scoring method	1.43	3.02

it can be seen in this chart that the scoring system is still not affected and the reported breathing rate by the scoring method is able to select the best performing algorithm dynamically. This fact is further emphasized by the error rates shown in Table 4.14.

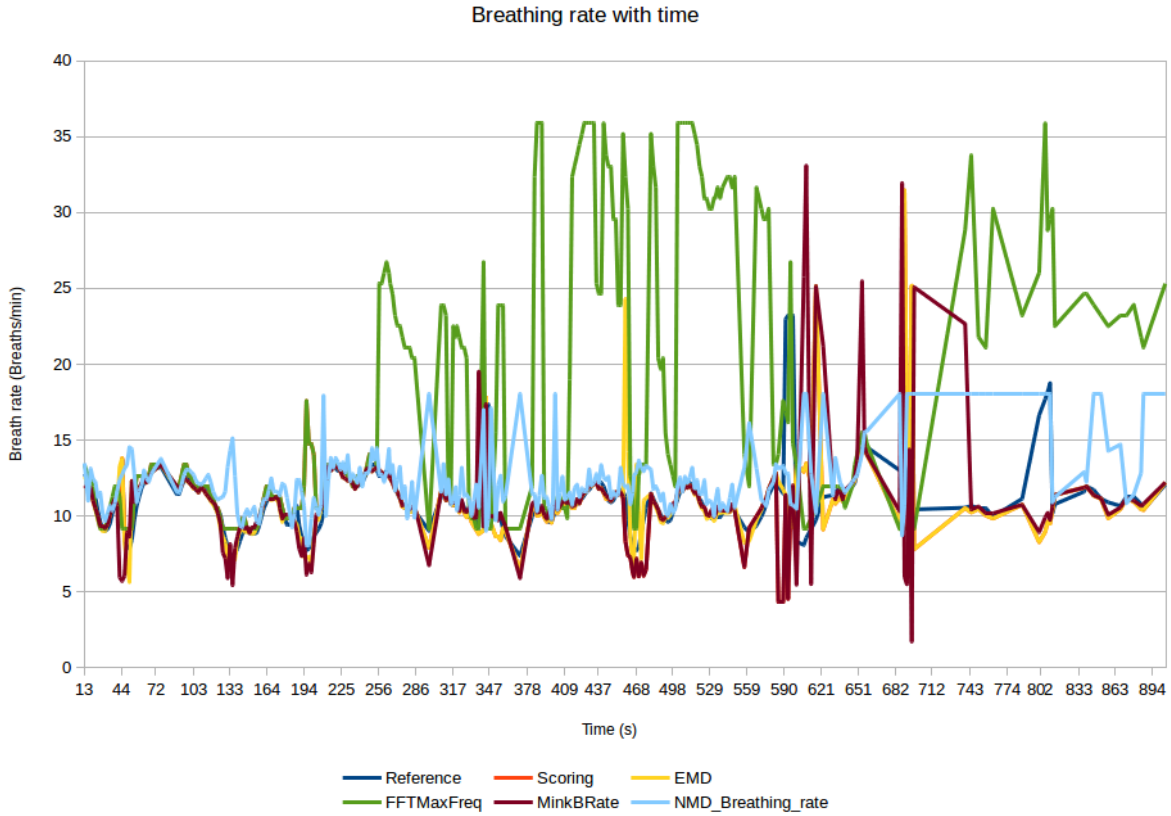


Figure 4.21: Breathing rate tracked through time

4.3.3.3 Subject 03

Figures 4.22, 4.23 and Tables 4.15, 4.16 shows two test cases with subject 03 and as can be seen in these figures and tables, the system is able to track the breathing rate with very minimal error.

The next section (Section 4.3.3.4 summarizes all the results from all 9 subjects studied and compares the performance of all evaluated algorithms.

Table 4.15: Mean error and standard deviation of breathing rate estimation for subject 03 (Units are in breaths/minute) for the test case corresponding to Figure 4.22

Algorithm name	Mean Absolute Error	Standard deviation of Absolute Error
EMD method	1.47	2.90
FFT Max.Freq. Method	0.65	0.29
Minkowski distance method	1.17	2.25
NMD method	2.20	1.88
Scoring method	0.86	0.68

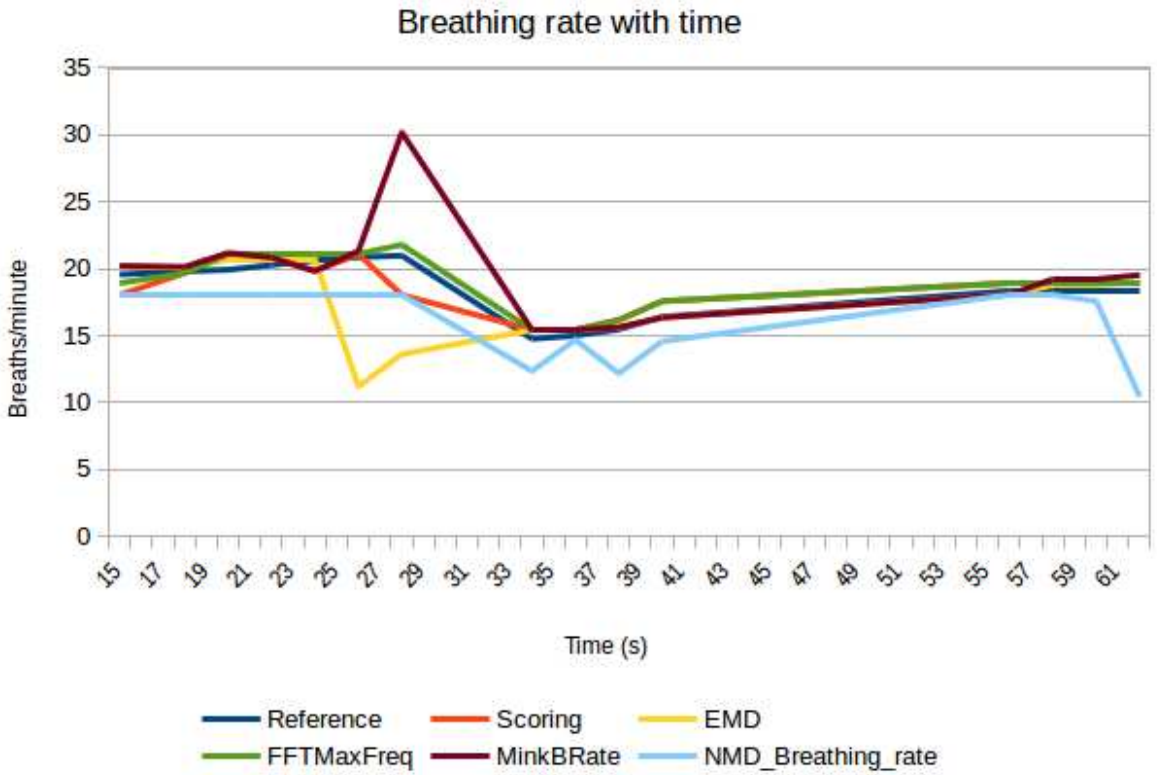


Figure 4.22: Breathing rate tracked through time

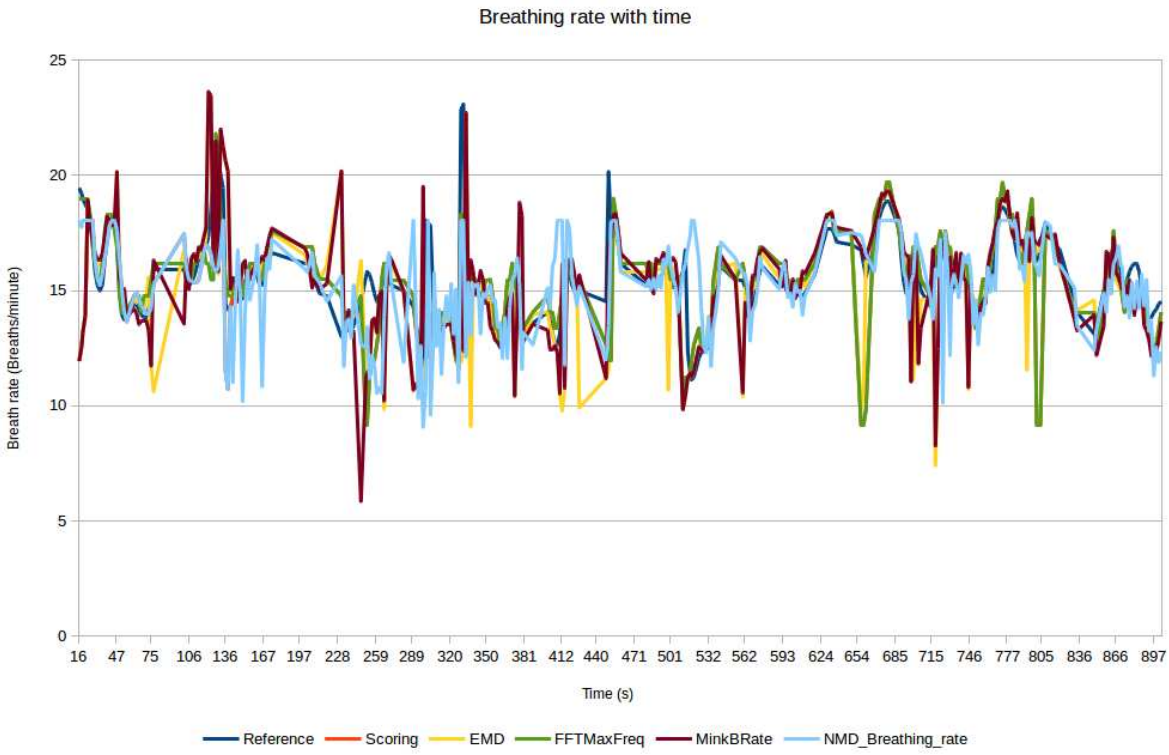


Figure 4.23: Breathing rate tracked through time

Table 4.16: Mean error and standard deviation of breathing rate estimation for subject 03 (Units are in breaths/minute) for the test case corresponding to Figure 4.23

Algorithm name	Mean Absolute Error	Standard deviation of Absolute Error
EMD method	1.28	1.93
FFT Max.Freq. Method	0.94	1.42
Minkowski distance method	1.27	1.75
NMD method	1.16	1.43
Scoring method	1.05	1.57

4.3.3.4 Summary for all subjects

This subsection contains the summary of results for selected best performing algorithms for all the subjects. As seen on Table B.31 on Appendix B.3, the system is able to estimate breathing rate with very minimal errors. It can also be seen that a single algorithm can not be chosen as the best performing algorithm as the performance of each algorithm varies within each test-case. However, the scoring method which selects the best performing algorithm dynamically is able to minimize the error in most of the cases. Even in cases when the performance of the scoring algorithm is worse than the best performing algorithm in that case, the scoring algorithm has mean error rates very close to the best performing algorithm. This gives the system more reliability than just using one algorithm as even the best performing algorithm in one case can be performing badly in another case. As it can be seen from Table B.31, the system is able to accurately estimate the breathing rate for all the subjects without any significant error for any particular subject. The scoring method is able to keep the mean error rate less than 1.03 breaths per minute in all the subjects. As explained on Figure 4.19, the system is able to estimate breathing rate even in the presence of small random body movements. Large body movements which inhibit the radars' ability to estimate a breathing rate are detected by the algorithm in Section 3.1.2, and the subjects were asked to stay stationary during the experiments. However, most of the subjects did small body movements as the tests were conducted for long periods of time (15 minutes), the subjects always showed small body movements as shown in Figure 4.19. However, it can be seen that the accuracy of breathing rate estimation is not affected adversely by observing the mean error rate and the standard deviation of the absolute error showing very small values.

Table 4.17 shows the mean absolute error of each algorithm for all 9 subjects and as seen on this table, the novel algorithm described in Section 3.2.2 (Named as 'EMD algorithm' in the

Table 4.17: Mean absolute error for all 9 subjects (Units are in breaths per minute)

Algorithm	Mean absolute error (Breaths/min)
EMD algorithm	0.79
FFTMaxFreq method	2.03
MinkBRate	2.18
NMD method	2.12
Scoring method	0.78

Table 4.18: Total number of times each algorithm generated best or second best results

	Total times with best results	Total times with second best results
EMD algorithm	4	4
FFTMaxFreq method	1	2
MinkBRate	2	1
NMD method	2	2

aforementioned table) shows the lowest mean error rate out of all algorithms and as it can be seen on this table, the scoring method to choose the best algorithm dynamically (described in Section 3.2.4) is able to further minimize the error in breathing rate estimation by dynamically selecting the best performing algorithm to report breathing rate. Individual results corresponding to each subject and each algorithm can be found in Appendix B.3.

Table 4.18 shows the total number of times each algorithm provided the best and second best results out of the 9 total subjects. As seen on this table, the EMD based algorithm in Section 3.2.2 provided the best results 4 times out of 9 and even in the cases when it did not provide the best result, it provided the second best result in another 4 out of 9 cases. As such, the EMD method provided the best or second best results in 8 out of the 9 total cases.

Chapter 5

Conclusion

Even though radar based contactless breathing rate estimation has been explored since the 1960s, the use of radars for this purpose is still not popular nor used for continuously monitoring people in real-world situations. The main goal of this research was to evaluate the existing methods/algorithms for breathing rate estimation and to evaluate the possibility of using these algorithms in a real-world situation. As the accuracy of most of the current algorithms were observed to be not satisfactory in real-world situations, a novel algorithm was developed to improve the breathing rate estimation in such situations. As many of the algorithms used in this field only work on specific situations and tend to fail intermittently when they are used in real-world situations, an algorithm was developed to score each breathing rate estimation algorithm dynamically and select the best algorithm automatically depending on the consistency of their results. The final system developed for this research consists of the novel breathing rate estimation algorithm along with several other existing algorithms estimating the breathing rate in parallel and the scoring system selecting the best performing algorithm dynamically. However, it should be noted that since the scoring system to select the best performing algorithm does not have any reference signal to evaluate the algorithms, the system selects the best performing algorithm based on their relative performance with each other and their individual performance through time. Hence, if all or a high majority of the algorithms used in this system are giving erroneous results, the scoring system is not able to select a good estimate for the breathing rate of the subject.

Most of the work in this field also focus mainly on test-cases with the subject being stationary and since in a real-world situation, the subject can be walking around a room or doing random

body movements which would interfere with capturing the breathing signal using a radar, in a real-world application, these movements should be automatically identified and the breathing signal or the breathing rate should not be reported in such situations as the estimates can be erroneous in such cases. A system was developed to identify the presence of a breathing-like signal and this enabled the breathing rate estimation to be performed only when a breathing signal is present in the radars' range. This system is critical in applying this solution in real-world situations as trying to estimate breathing rate when a breathing signal is not present can introduce problems as the system would be reporting erroneous results.

Another requirement in such a system is to test it under many different situations. However, if this testing and data analysis was done manually, the researcher(s) would have to spend a lot of time just collecting, labelling and processing the data for each test-case manually. Hence, an automated system was developed that is able to collect radar data simultaneously with a reference sensor and then process the data and generate reports about the accuracy of all the tested algorithms, comparing them with the reference data. This system enabled the development stage to be fast-tracked as iterative developments were faster as the testing of the complete system was automatic.

The final system developed as a result of this work is a Python™ application that performs breathing rate estimation and related tasks in the following order;

1. Locate a person in the radars' range.
2. Check for the presence of a breathing-like signal
3. Estimate breathing rate using a novel breathing rate estimation algorithm and many other existing algorithms simultaneously.
4. Select the best performing algorithm dynamically.
5. Send the results to a local GUI and a cloud server.

This system was also developed in a modular fashion and is compatible with both Python™ and Matlab® based algorithms for breathing rate estimation. New algorithms can be incorporated into this system by simply writing a Matlab or Python script and the same scripts can be

incorporated into the report generation frameworks as well without any modification of the code, and this allows for expansion of this work with minimal effort.

5.1 Limitations and future work

Even-though the system introduced in Section 3.1.2 is able to detect the presence of a breathing like signal, the thresholds used in the auto-correlation based approach to determine the presence of a breathing like signal needs to be adjusted if the requirement arises to detect breathing rates higher than normal breathing rates for healthy humans. The thresholds used in this work can identify breathing signals up to 30 breaths per minute and a threshold that is too low could make the system produce a lot of false positives on breathing-like signal detection. It is assumed that a person having a breathing rate higher than 30 breaths per minute for prolonged periods of time would be suffering from a respiratory disease that would be detected by a doctor or a certified medical device.

Distinguishing between other movements generated by sources such as fans, moving water, electric motors, etc. and human breathing signals and the effect on the breathing rate estimation in the presence of such devices which follow periodic or random movement patterns have not been explored in this work. Studying the effects of other sources of physical movement on the area being monitored by radar can be studied as part of a future study.

Even-though it is known that UWB radar is capable of penetrating solid obstacles such as partition walls etc. which are not made of metal, the effect of such an obstacle between the radar and the subject has not been explored in this research. Exploring the effect of obstacles such as partition walls, thick blankets, etc. between the radar and the subject can also be studied as part of a future study.

Some other possible future work based on this work are as follows:

1. The performance of the algorithms presented in this research will be tested under realistic conditions in a real elder care facility environment, with both healthy elders and elders with conditions affecting their respiration. A three month long period of uncontrolled data collection is planned to collect data from seniors in an elder care facility in Ottawa and the data-collection systems described in this thesis will be used for this task and the data

will be analyzed while the data is being collected and the results will automatically be sent to a cloud server (provided that an internet connection is available). The raw data will also be saved locally and analyzed separately by researchers. All the software components required for data collection and local analysis were developed during this thesis work.

2. The automatic algorithm selection system described in Section 3.2.4 can be improved by incorporating machine learning algorithms that can understand which algorithms would perform well in different patient/posture/activity level combinations. A possible system could be trained in real-time using reference data from a breathing belt (The data-collection system described in Section 3.3 could be extended for this purpose).

3. Additional information about the subject such as accurate posture and activity level classifications along with fall detection can be done by synchronizing the radar data with a 3D camera. Initial work on this has already been carried out as shown in Appendix C.2.

4. Moving block bootstrap[1] based confidence interval estimation can be incorporated into the system by running the breathing rate estimation thread described in Section 3.6. Currently this thread runs on moving blocks of time in a deterministic manner (one window after another with each window constructed by advancing time). This process could be randomized and breathing rates could be estimated for multiple blocks of time in a parallel manner by running multiple instances of the same Python™ class developed for breathing rate estimation. These confidence intervals can then be used for evaluating the breathing rate estimation algorithms (Provided that enough computing resources are available to perform block bootstrap confidence intervals for each algorithm) and select the best performing algorithm dynamically.

[1] https://en.wikipedia.org/wiki/Bootstrapping_(statistics)

Bibliography

[1] J. C. Lin, “Microwave sensing of physiological movement and volume change: A review,” *Bioelectromagnetics*, vol. 13, pp. 557–565, Jan. 1992.

[2] Z. Baird, I. Gunasekara, M. Bolic, and S. Rajan, “Principal component analysis-based occupancy detection with ultra wideband radar,” in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1573–1576, Aug. 2017.

[3] C. Wolff, “Radar basics.” "Available at: http://www.radartutorial.eu/, accessed 2017-12-01".

[4] A. Wilzeck, M. Perez Guirao, and E. Dimtrow, *UWB Technology and Regulation*. wiseSense GmbH.

[5] C. Nguyen and M. Miao, *Fundamentals of UWB Impulse Systems*, pp. 7–24. Cham: Springer International Publishing, 2017.

[6] “WiFi Channels | WiFi Frequency Bands List | Radio-Electronics.com.” "Available at: http://www.radio-electronics.com/info/wireless/wi-fi/80211-channels-number-frequencies-bandwidth.php, accessed 2017-12-01".

[7] Time Domain, Huntsville, Alabama, *Time Domain’s Ultra Wideband (UWB), Definition and Advantages*, 2012.

[8] H. Langen, *Ultra-Wideband Radar Simulator for classifying Humans and Animals based on Micro-Doppler Signatures*. PhD thesis, Department of Electronics and Telecommunications, Norwegian University of Science and Technology, 2016.

[9] M. I. Skolnik, *Introduction to Radar Systems*. McGrawHill Education, 3rd edition, 2002.

[10] Xethru by Novelda, *X4A02 Datasheet*. https://www.xethru.com/community/resources/x4a02-antenna-board-datasheet.87/, accessed 2017-12-01.

[11] Novelda AS, *Novelda 6.0 – 8.5 GHz Sinuous Antenna Technical Manual*. https://www.xethru.com/press-room/detail/wideband-antenna-for-novelda-nanoscale0-impulse-radar.html, accessed 2017-12-01.

[12] Xethru by Novelda, *Xetrhru X2 Impulse Radar Transceiver Datasheet*. https://www.xethru.com/community/resources/x2-datasheet.24/, accessed 2017-12-01.

[13] M. Piórek and W. Winiecki, "On calibration and parametrization of low power ultrawideband radar for close range detection of human body and bodily functions," in *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2, pp. 639–645, Sept. 2015.

[14] Xethru by Novelda, *XeThru Sensor Emissions*, 2016.

[15] P. K. Verma, A. N. Gaikwad, D. Singh, and M. J. Nigam, "Analysis of Clutter Reduction Techniques for through Wall Imaging in UWB Range," *Progress In Electromagnetics Research B*, vol. 17, pp. 29–48, 2009.

[16] S. Singh, Q. Liang, D. Chen, and L. Sheng, "Sense through wall human detection using UWB radar," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, p. 20, Dec. 2011.

[17] M. H. E. Mabrouk, *Signal Processing of Uwb Radar Signals for Human Detection Behind Walls*. PhD thesis, School of Electrical Engineering and Computer Science, University of Ottawa, 2015.

[18] F. Khan, J. W. Choi, and S. H. Cho, "Vital sign monitoring of a non-stationary human through IR-UWB radar," in *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*, pp. 511–514, Sept. 2014.

[19] F. Khan and S. H. Cho, "A Detailed Algorithm for Vital Sign Monitoring of a Stationary/Non-Stationary Human through IR-UWB Radar," *Sensors*, vol. 17, p. 290, Feb. 2017.

[20] H. Guo, "A method of life signal identification based on BP neural network," in *2011 4th International Congress on Image and Signal Processing*, vol. 1, pp. 22–25, Oct. 2011.

[21] Y. Liao, *Phase and Frequency Estimation: High-Accuracy and Low-Complexity Techniques*. PhD thesis, Worcester Polytechnic Institute, 2011.

[22] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 476–480, May 1999.

[23] D. Ruan, J. A. Fessler, and J. M. Balter, "Mean position tracking of respiratory motion," *Medical Physics*, vol. 35, no. 2, pp. 782–792, 2008.

[24] C. Li, *Non-contract Estimation of Respiration and Heartbeat Rate using Ultra-Wideband Signals*. PhD thesis, Virginia Polytechnic Institute and State University, 2008.

[25] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, pp. 70–73, Jun 1967.

[26] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*, vol. 454, pp. 903–995, The Royal Society, 1998.

[27] Q. Jian, J. Yang, Y. Yu, P. Björkholm, and T. McKelvey, "Detection of breathing and heartbeat by using a simple UWB radar system," in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, pp. 3078–3081, Apr. 2014.

[28] J. Sun and M. Li, "Life detection and location methods using UWB impulse radar in a coal mine," *Mining Science and Technology (China)*, vol. 21, pp. 687–691, Sept. 2011.

[29] R. M. Narayanan, "Through wall radar imaging using UWB noise waveforms," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5185–5188, Mar. 2008.

[30] D. Labate, F. L. Foresta, G. Occhiuto, F. C. Morabito, A. Lay-Ekuakille, and P. Vergallo, "Empirical Mode Decomposition vs. Wavelet Decomposition for the Extraction of Respiratory Signal From Single-Channel ECG: A Comparison," *IEEE Sensors Journal*, vol. 13, pp. 2666–2674, July 2013.

[31] X. Zhang, "Characterizing performance of the radar system for breathing and heart rate estimation in real-life conditions," Master's thesis, School of Electrical Engineering and Computer Science, University of Ottawa, 2017.

[32] R. Ravichandran, E. Saba, K. Y. Chen, M. Goel, S. Gupta, and S. N. Patel, "Wibreathe: Estimating respiration rate using wireless signals in natural settings in the home," in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 131–139, March 2015.

[33] O. Kaltiokallio, H. Yiğitler, R. Jäntti, and N. Patwari, "Non-invasive respiration rate monitoring using a single cots tx-rx pair," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pp. 59–69, April 2014.

[34] Y. Li, X. Jing, H. Lv, and J. Wang, "Analysis of characteristics of two close stationary human targets detected by impulse radio UWB radar," vol. 126, pp. 429–447.

[35] P. J. J. Luukko, J. Helske, and E. Räsänen, "Introducing libeemd: a program package for performing the ensemble empirical mode decomposition," *Computational Statistics*, vol. 31, pp. 545–557, June 2016.

[36] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Advances in Adaptive Data Analysis*, vol. 01, pp. 1–41, Jan. 2009.

[37] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4144–4147, May 2011.

[38] D. Iatsenko, P. V. E. McClintock, and A. Stefanovska, "Nonlinear Mode Decomposition: a new noise-robust, adaptive decomposition method," *Physical Review E*, vol. 92, Sept. 2015. arXiv: 1207.5567.

[39] Diksha, "Difference between Process and Thread," Dec. 2013. Available at: http://www.differencebetween.info/difference-between-process-and-thread, accessed 2017-12-01.

[40] Z. J. Baird, "Human activity and posture classification using single non-contact radar sensor," Master's thesis, Carleton University, Ottawa, Ontario, 2017.

Appendix A

Ethics approvals for experiments

This section contains the ethics approvals granted by the University of Ottawa Office of Research Ethics and Integrity for the experiments conducted for this research.

File Number: H06-17-23 Date (mm/dd/yyyy): 11/24/2017

Université d'Ottawa University of Ottawa
Bureau d'éthique et d'intégrité de la recherche Office of Research Ethics and Integrity

Ethics Approval Notice

Health Sciences and Science REB

Principal Investigator / Supervisor / Co-investigator(s) / Student(s)

First Name	Last Name	Affiliation	Role
Miodrag	Bolic	Engineering / Computer Science	Principal Investigator
Sreeraman	Rajan	Engineering / Electrical Engineering	Co-investigator
Zachary	Baird	Engineering / Computer Engineering	Research Assistant
Isuru	Gunasekara	Engineering / Computer Engineering	Research Assistant
Britto	Martial	Engineering / Computer Engineering	Research Assistant
Hamza	Qassoud	Engineering / Computer Engineering	Research Assistant

File Number: H06-17-23

Type of Project: Professor

Title: Human physiological monitoring using UWB radar and 3D depth sensor

Approval Date (mm/dd/yyyy)	Expiry Date (mm/dd/yyyy)	Approval Type
11/24/2017	11/23/2018	Approval

Special Conditions / Comments:
N/A

550, rue Cumberland, pièce 154 550 Cumberland Street, room 154
Ottawa (Ontario) K1N 6N5 Canada Ottawa, Ontario K1N 6N5 Canada
(613) 562-5387 • Téléc./Fax (613) 562-5338
www.recherche.uottawa.ca/deontologie/ www.research.uottawa.ca/ethics/

File Number: H06-17-23 **Date (mm/dd/yyyy):** 11/24/2017

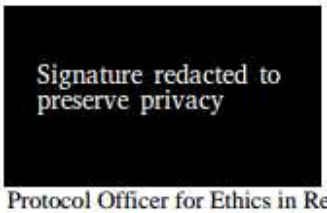
Université d'Ottawa University of Ottawa
Bureau d'éthique et d'intégrité de la recherche Office of Research Ethics and Integrity

This is to confirm that the University of Ottawa Research Ethics Board identified above, which operates in accordance with the Tri-Council Policy Statement (2010) and other applicable laws and regulations in Ontario, has examined and approved the ethics application for the above named research project. Ethics approval is valid for the period indicated above and subject to the conditions listed in the section entitled "Special Conditions / Comments".

During the course of the project, the protocol may not be modified without prior written approval from the REB except when necessary to remove participants from immediate endangerment or when the modification(s) pertain to only administrative or logistical components of the project (e.g., change of telephone number). Investigators must also promptly alert the REB of any changes which increase the risk to participant(s), any changes which considerably affect the conduct of the project, all unanticipated and harmful events that occur, and new information that may negatively affect the conduct of the project and safety of the participant(s). Modifications to the project, including consent and recruitment documentation, should be submitted to the Ethics Office for approval using the "Modification to research project" form available at: https://research.uottawa.ca/ethics/forms.

Please submit an annual report to the Ethics Office four weeks before the above-referenced expiry date to request a renewal of this ethics approval. To close the file, a final report must be submitted. These documents can be found at: https://research.uottawa.ca/ethics/forms.

If you have any questions, please do not hesitate to contact the Ethics Office at extension 5387 or by e-mail at: ethics@uOttawa.ca.



Protocol Officer for Ethics in Research
For Daniel Lagarec, Chair of the Health Sciences and Sciences REB

550, rue Cumberland, pièce 154 550 Cumberland Street, room 154
Ottawa (Ontario) K1N 6N5 Canada Ottawa, Ontario K1N 6N5 Canada
(613) 562-5387 • Téléc./Fax (613) 562-5338
www.recherche.uottawa.ca/deontologie/ www.research.uottawa.ca/ethics/

Université d'Ottawa University of Ottawa
Bureau d'éthique et d'intégrité de la recherche Office of Research Ethics and Integrity

Ethics Approval Notice

Health Sciences and Science REB

Principal Investigator / Supervisor / Co-investigator(s) / Student(s)

First Name	Last Name	Affiliation	Role
Miodrag	Bolic	Engineering / Computer Science	Principal Investigator
Sreeraman	Rajan	Engineering / Electrical Engineering	Co-Principal Investigator
Olufemi	Adeojo	Engineering / Others	Research Assistant
Zachary	Baird	Engineering / Computer Engineering	Research Assistant
Gangesh	Beri	Engineering / Others	Research Assistant
Anwar	Fallatah	Engineering / Electrical Engineering	Research Assistant
Isuru	Gunasekara	Engineering / Computer Engineering	Research Assistant
Samuel	Hadwen	Others / Others	Research Assistant
Britto	Martial	Engineering / Computer Engineering	Research Assistant
Alexander	May	Engineering / Computer Engineering	Research Assistant
Hamza	Qassoud	Engineering / Computer Engineering	Research Assistant
Munir	Tarar	Others / Others	Research Assistant

File Number: H08-17-27

Type of Project: Professor

Title: Contactless Detection of Breathing and Falls of Humans using UWB RADAR and 3D Motion Sensor

Approval Date (mm/dd/yyyy)	Expiry Date (mm/dd/yyyy)	Approval Type
10/27/2017	10/26/2018	Approval

550, rue Cumberland, pièce 154 550 Cumberland Street, room 154
Ottawa (Ontario) K1N 6N5 Canada Ottawa, Ontario K1N 6N5 Canada
(613) 562-5387 • Téléc./Fax (613) 562-5338
www.recherche.uottawa.ca/deontologie/ www.research.uottawa.ca/ethics/

Université d'Ottawa University of Ottawa
Bureau d'éthique et d'intégrité de la recherche Office of Research Ethics and Integrity

This is to confirm that the University of Ottawa Research Ethics Board identified above, which operates in accordance with the Tri-Council Policy Statement (2010) and other applicable laws and regulations in Ontario, has examined and approved the ethics application for the above named research project. Ethics approval is valid for the period indicated above and subject to the conditions listed in the section entitled "Special Conditions / Comments".

During the course of the project, the protocol may not be modified without prior written approval from the REB except when necessary to remove participants from immediate endangerment or when the modification(s) pertain to only administrative or logistical components of the project (e.g., change of telephone number). Investigators must also promptly alert the REB of any changes which increase the risk to participant(s), any changes which considerably affect the conduct of the project, all unanticipated and harmful events that occur, and new information that may negatively affect the conduct of the project and safety of the participant(s). Modifications to the project, including consent and recruitment documentation, should be submitted to the Ethics Office for approval using the "Modification to research project" form available at: http://research.uottawa.ca/ethics/submissions-and-reviews.

Please submit an annual report to the Ethics Office four weeks before the above-referenced expiry date to request a renewal of this ethics approval. To close the file, a final report must be submitted. These documents can be found at: http://research.uottawa.ca/ethics/submissions-and-reviews.

If you have any questions, please do not hesitate to contact the Ethics Office at extension 5387 or by e-mail at: ethics@uOttawa.ca.

Signature:

Signature redacted to preserve privacy

Protocol Officer for Ethics in Research
For Daniel Lagarec, Chair of the Health Sciences and Sciences REB

550, rue Cumberland, pièce 154 550 Cumberland Street, room 154
Ottawa (Ontario) K1N 6N5 Canada Ottawa, Ontario K1N 6N5 Canada
(613) 562-5387 • Téléc./Fax (613) 562-5338
www.recherche.uottawa.ca/deontologie/ www.research.uottawa.ca/ethics/

Appendix B

Detailed results

B.1 Breathing rate estimation results for stationary subject at multiple positions and orientations

This section contains the results of breathing rate estimation for the test cases corresponding to Section 4.3.2.

Table B.1: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P1 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	1.65	1.68
FFTMaxFreq method	0.77	0.47
MinkBRate method	3.07	4.05
NMD method	1.24	1.98
Scoring method	1.11	1.58

Table B.2: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P2 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	0.59	0.27
FFTMaxFreq method	0.67	0.28
MinkBRate method	0.60	0.27
NMD method	0.67	1.60
Scoring method	0.44	0.38

Table B.3: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P3 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	1.17	1.43
FFTMaxFreq method	0.56	0.21
MinkBRate method	6.52	9.40
NMD method	0.73	1.15
Scoring method	0.51	0.22

Table B.4: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P4 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	1.30	1.21
FFTMaxFreq method	0.90	0.57
MinkBRate method	1.52	1.26
NMD method	1.26	1.75
Scoring method	0.92	0.65

Table B.5: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P1 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	3.59	3.23
FFTMaxFreq method	2.20	1.51
MinkBRate method	4.75	3.62
NMD method	2.33	2.00
Scoring method	2.24	1.98

Table B.6: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P2 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	4.09	2.71
FFTMaxFreq method	3.16	3.07
MinkBRate method	4.88	4.01
NMD method	2.42	1.43
Scoring method	2.23	1.37

Table B.7: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P3 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	4.23	2.24
FFTMaxFreq method	3.23	2.63
MinkBRate method	5.74	2.19
NMD method	2.53	2.04
Scoring method	3.06	2.11

Table B.8: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P4 with their front towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	2.74	2.22
FFTMaxFreq method	1.70	1.18
MinkBRate method	3.17	3.45
NMD method	2.67	2.15
Scoring method	1.72	1.26

Table B.9: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P1 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	1.10	0.77
FFTMaxFreq method	0.66	0.39
MinkBRate method	1.13	0.74
NMD method	1.28	2.06
Scoring method	0.64	0.42

Table B.10: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P2 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	1.31	1.68
FFTMaxFreq method	1.25	1.85
MinkBRate method	65.48	126.65
NMD method	3.27	1.76
Scoring method	1.68	1.84

Table B.11: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P3 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	2.64	2.15
FFTMaxFreq method	2.24	1.67
MinkBRate method	8.04	3.33
NMD method	1.93	1.72
Scoring method	3.99	3.58

Table B.12: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P4 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	2.84	2.22
FFTMaxFreq method	2.04	1.36
MinkBRate method	4.53	3.25
NMD method	2.24	1.40
Scoring method	2.04	1.36

Table B.13: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P1 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	4.34	3.75
FFTMaxFreq method	3.14	2.29
MinkBRate method	5.01	3.97
NMD method	2.06	1.39
Scoring method	3.28	2.78

Table B.14: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P2 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	3.42	3.99
FFTMaxFreq method	2.13	2.33
MinkBRate method	3.39	4.16
NMD method	1.82	2.11
Scoring method	1.91	2.49

Table B.15: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P3 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	7.12	3.55
FFTMaxFreq method	5.66	3.25
MinkBRate method	5.17	5.70
NMD method	1.93	2.17
Scoring method	6.11	3.50

Table B.16: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P4 with their back towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	3.16	4.35
FFTMaxFreq method	0.75	0.65
MinkBRate method	7.69	8.56
NMD method	1.31	1.91
Scoring method	0.68	0.66

Table B.17: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P1 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	3.95	5.26
FFTMaxFreq method	1.83	4.12
MinkBRate method	4.16	5.22
NMD method	1.11	1.44
Scoring method	1.99	2.81

Table B.18: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P2 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	0.81	1.71
FFTMaxFreq method	0.96	1.67
MinkBRate method	0.79	1.64
NMD method	1.67	1.57
Scoring method	0.80	1.63

Table B.19: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P3 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	1.48	3.89
FFTMaxFreq method	0.53	0.91
MinkBRate method	2.95	5.05
NMD method	0.91	1.37
Scoring method	0.45	0.47

Table B.20: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject sitting at point P4 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	5.98	4.51
FFTMaxFreq method	6.19	5.14
MinkBRate method	6.57	4.30
NMD method	3.34	1.73
Scoring method	4.39	3.74

Table B.21: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P1 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	2.44	3.08
FFTMaxFreq method	2.78	3.37
MinkBRate method	2.90	4.41
NMD method	1.22	2.18
Scoring method	2.68	3.40

Table B.22: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P2 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	3.85	5.07
FFTMaxFreq method	2.01	2.38
MinkBRate method	4.21	3.87
NMD method	2.33	1.99
Scoring method	1.76	2.09

Table B.23: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P3 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	5.91	3.05
FFTMaxFreq method	4.07	2.39
MinkBRate method	10.13	4.91
NMD method	2.58	2.35
Scoring method	4.13	2.98

Table B.24: Mean absolute error and standard deviation of absolute error in breathing rate estimation with subject standing at point P4 with their left side towards the radar. (Units are in breaths/minute)

Algorithm	Mean Absolute Error(Br/min)	Standard deviation of Absolute Error
EMD method	3.44	5.21
FFTMaxFreq method	0.63	0.47
MinkBRate method	5.59	8.68
NMD method	0.82	1.09
Scoring method	0.86	1.13

B.2 Breathing rate estimation results from miscellaneous algorithms not used in the final system

This section contains the results for Section 4.3.2.1. The following abbreviations are used in this subsection to describe the test-cases (Subject's posture, orientation, location)

- FR: Subject is facing the radar (Their front is towards the radar).
- BA: Subject has their back towards the radar.
- SL: Subject has their left side towards the radar.
- P1,P2,P3,P4: Subjects' location. A figure showing the locations corresponding to P1,P2,P3 and P4 can be found in Figure 4.1.

Table B.25: Mean absolute error in breaths per minute for all test cases with subject sitting in front, facing the radar

Algorithm	FR-P1	FR-P2	FR-P3	FR-P4
Shannon energy based algorithm	9.20	3.62	2.49	5.91
Notch filter based algorithm	7.16	1.60	1.32	3.33
Ellipse fitting algorithm	9.43	3.44	4.63	7.38
Wavelet transform based method	5.71	0.91	1.51	2.05
Welch periodogram	2.87	1.77	1.51	2.33
Rectangular moving window algorithm	8.86	3.61	2.20	5.74
Sum squared algorithm	9.65	2.50	4.04	5.14

Table B.26: Mean absolute error in breaths per minute for all test cases with subject standing in front, facing the radar

Algorithm	FR-P1	FR-P2	FR-P3	FR-P4
Shannon energy based algorithm	9.30	7.44	8.60	8.95
Notch filter based algorithm	7.40	6.66	7.57	5.84
Ellipse fitting algorithm	9.22	8.45	9.70	8.93
Wavelet transform based method	6.51	9.01	7.92	7.36
Welch periodogram	2.95	4.25	3.58	2.80
Rectangular moving window algorithm	9.28	7.85	8.59	8.94
Sum squared algorithm	9.09	7.54	8.54	8.99

Table B.27: Mean absolute error in breaths per minute for all test cases with subject sitting in front of radar with their back towards the radar

Algorithm	BA-P1	BA-P2	BA-P3	BA-P4
Shannon energy based algorithm	5.11	10.31	7.47	6.69
Notch filter based algorithm	2.50	10.21	6.03	4.38
Ellipse fitting algorithm	5.79	10.43	6.18	8.18
Wavelet transform based method	3.14	6.36	9.31	9.67
Welch periodogram	2.38	7.73	2.18	3.56
Rectangular moving window algorithm	6.13	10.25	7.27	6.66
Sum squared algorithm	4.86	10.35	7.33	6.81

Table B.28: Mean absolute error in breaths per minute for all test cases with subject standing in front of radar with their back towards the radar

Algorithm	BA-P1	BA-P2	BA-P3	BA-P4
Shannon energy based algorithm	7.69	7.64	10.06	4.43
Notch filter based algorithm	6.18	5.56	9.79	3.51
Ellipse fitting algorithm	9.64	8.50	11.26	6.15
Wavelet transform based method	9.03	9.25	10.06	5.16
Welch periodogram	3.39	2.51	3.57	3.21
Rectangular moving window algorithm	7.96	7.48	9.75	4.98
Sum squared algorithm	7.61	7.22	10.04	4.44

Table B.29: Mean absolute error in breaths per minute for all test cases with subject sitting in front of radar with their left side towards the radar

Algorithm	SL-P1	SL-P2	SL-P3	SL-P4
Shannon energy based algorithm	6.39	5.29	7.18	7.12
Notch filter based algorithm	3.24	5.30	2.53	6.25
Ellipse fitting algorithm	6.28	5.17	8.13	7.20
Wavelet transform based method	4.40	1.81	5.09	9.36
Welch periodogram	1.34	5.00	3.13	5.01
Rectangular moving window algorithm	6.47	5.17	5.86	6.96
Sum squared algorithm	6.43	5.39	7.21	7.15

Table B.30: Mean absolute error in breaths per minute for all test cases with subject standing in front of radar with their left side towards the radar

Algorithm	SL-P1	SL-P2	SL-P3	SL-P4
Shannon energy based algorithm	10.02	8.46	6.55	4.95
Notch filter based algorithm	5.95	7.85	6.45	3.79
Ellipse fitting algorithm	10.43	8.86	7.95	6.54
Wavelet transform based method	7.62	5.93	8.37	7.09
Welch periodogram	2.78	3.07	4.33	2.29
Rectangular moving window algorithm	9.95	8.15	6.55	5.37
Sum squared algorithm	9.65	8.46	6.37	4.99

B.3 Breathing rate estimation results for comprehensive study with multiple subjects

This section contains results corresponding to Section 4.3.3.

Table B.31: Mean absolute error and standard deviation of absolute error for subjects 04 to 10 (Units are in breaths per minute)

	Mean absolute error (Br/min)	**Standard deviation of absolute error**
Subject 04		
EMD algorithm	0.39	0.22
FFTMaxFreq method	1.97	3.06
MinkBRate	3.18	5.10
NMD method	1.92	2.60
Scoring method	0.42	0.28
Subject 05		
EMD algorithm	0.22	0.21
FFTMaxFreq method	0.53	0.22
MinkBRate	0.12	0.12
NMD method	0.63	1.37
Scoring method	0.13	0.15
Subject 06		
EMD algorithm	0.91	1.11
FFTMaxFreq method	2.17	3.00
MinkBRate	1.63	2.97
NMD method	0.69	0.88
Scoring method	0.71	0.95
Subject 07		
EMD algorithm	0.89	2.93
FFTMaxFreq method	0.97	2.90
MinkBRate	1.03	2.84
NMD method	0.72	1.10
Scoring method	1.03	2.84
Subject 08		
EMD algorithm	0.47	0.66
FFTMaxFreq method	0.97	1.45
MinkBRate	7.24	7.00
NMD method	6.06	2.15
Scoring method	0.47	0.65
Subject 9		
EMD algorithm	0.58	0.97
FFTMaxFreq method	2.06	4.46
MinkBRate	0.54	0.81
NMD method	1.33	1.72
Scoring method	0.61	1.02

Appendix C

Other things done during this research

This Appendix contains details about other miscellaneous things done during this thesis work.

C.1 Startup Garage program organized by the University of Ottawa and Invest Ottawa

This research work, along with a business development and commercialization plan developed by Dr. Miodrag Bolic, Dr. Munir Tarar, Samuel Hadwen and Dr. Sreeraman Rajan won CAD$16,500 for commercialization of this work along with access to space and resources at the Innovation Centre at Bayview Yards (ICBY) for commercializing a contactless system to monitor elders at elder-care facilities through a competition organized by the University of Ottawa's Innovation Support Services. A complete end to end system to get radar data from the radar, process them and send the data to the cloud was demonstrated during several occasions.

C.2 Synchronization of the radar data with a 3D depth sensor

The radar data was synchronized with a 3D depth sensor and presented on a 3D frame along with breathing rate estimation as a possible future work containing a radar sensor for breathing rate

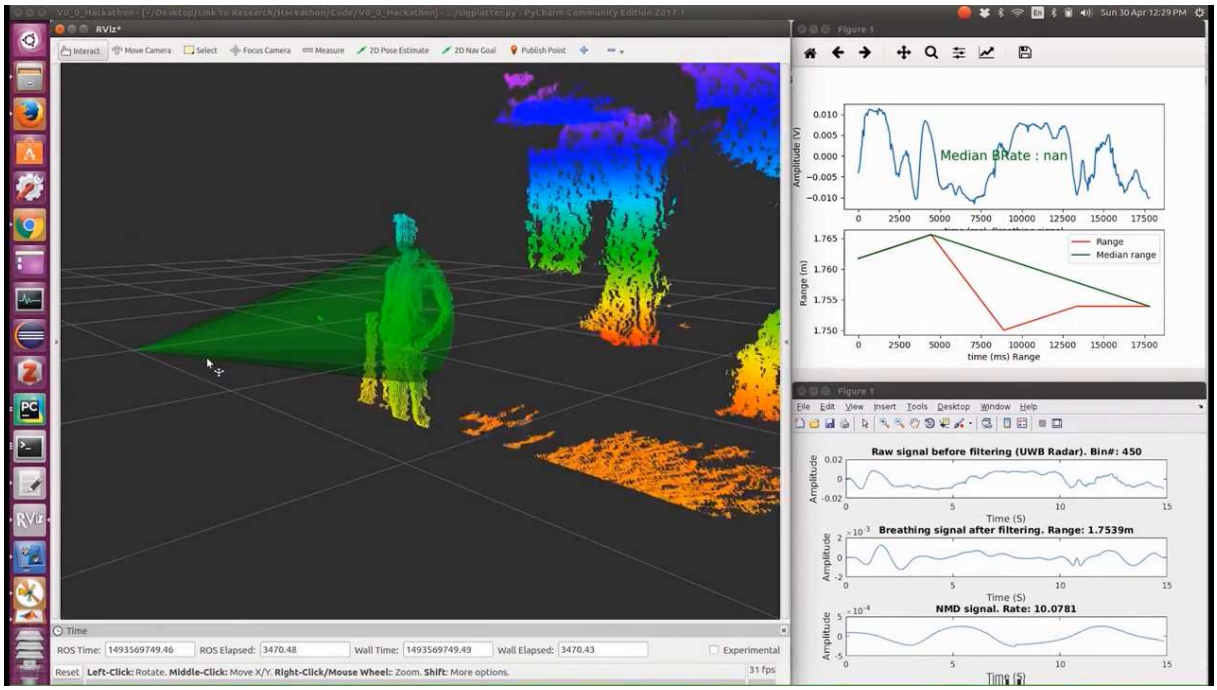


Figure C.1: Screen capture of the 3D video with radar data overlay. (radar based range estimation is overlay-ed as a green cone on the 3d video)

detection and a 3D camera for posture and activity classification along with detecting falls. A screen capture of the system working together can be seen in Figure C.1.