FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Jaroslava Schovancová

Remote sensoric detection of human vital signs

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis:	RNDr. David Obdržálek, Ph. D.
Study programme:	Computer Science
Study branch:	Artificial Intelligence

Prague 2024

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

To my family, my heroes.

Title: Remote sensoric detection of human vital signs

Author: Mgr. Jaroslava Schovancová, Ph. D.

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. David Obdržálek, Ph. D.

Abstract: In this work we explore an automated remote sensoric detection of human vital signs, using a frequency-modulated continuous-wave (FMCW) radar. It can have a positive impact on CERN LHC Operations, potentially assisting the remote Fire & Rescue service Operators in assessing health status of personnel in need of their assistance.

We explore limitations of such a detection in industrial settings, focusing particularly on the respiratory rate. We have investigated the detection gain & success w.r.t. different breathing patterns, human subject's body position and distance from the sensor, detection from different locations on the body, different subject's clothing, and different environmental conditions, including obstacles in the line of beam. We discuss different data analysis approaches.

With the available sensor, we observed that the best detection capabilities were with detection from the human subject's thorax independent on the body position, and on expected work clothing, with rather a not shallow and quick breathing pattern, in distance of up to several meters and no physical obstacles in the line of beam. We described several parametric models for respiratory rate detection with the radar. We described several parametric models for peak detection in chest belt and distance laser data.

Keywords: vital signs detection; radar; robotics medical applications

Title: Vzdálená senzorická detekce lidských životních funkcí

Author: Mgr. Jaroslava Schovancová, Ph. D.

Department: Katedra teoretické informatiky a matematické logiky

Supervisor: RNDr. David Obdržálek, Ph. D.

Abstract: V této práci zkoumáme automatizovanou vzdálenou senzorickou detekci lidských životních funkcí pomocí radaru s frekvenčně modulovanou kontinuální vlnou (FMCW). To může mít pozitivní dopad na provoz LHC v CERNu, a potenciálně pomoci vzdáleným operátorům služby hasičského a záchranného sboru při hodnocení zdravotního stavu osob, které potřebují jejich pomoc.

Zkoumáme omezení takové detekce v průmyslových prostředích, se zaměřením zejména na dechovou frekvenci. Zkoumali jsme úspěšnost detekce v závislosti na různých dechových vzorech, poloze těla lidského subjektu a vzdálenosti od senzoru, detekci z různých míst na těle, různém oblečení subjektu a různých podmínkách prostředí, včetně překážek v linii paprsku. Diskutujeme různé přístupy k analýze dat.

S dostupným senzorem jsme pozorovali, že nejlepší detekční schopnosti byly při detekci z hrudníku lidského subjektu nezávisle na poloze těla a na očekávaném pracovním oblečení, s ne příliš mělkým a rychlým dechem, ve vzdálenosti až několika metrů a bez fyzických překážek v linii paprsku. Popsali jsme několik parametrických modelů pro detekci dechové frekvence radarem. Popsali jsme několik parametrických modelů pro detekci špiček v datech z hrudního pásu a laseru měřícího vzdálenosti.

Keywords: detekce životních funkcí; radar; robotické medicínské aplikace

Acknowledgments

I started working on this topic several years ago when the COVID-19 pandemic started which, unfortunately, is still around, with colleagues from the CERN's Mechatronics, Robotics and Operations Section in the Beams department. Although my journey with CERN has concluded, I truly enjoyed and learned a lot in these past 18 years, from colleagues in ATLAS Computing, communities of Small and Medium Experiments, WLCG and CERN IT. With these lines, with a great pleasure, I would like to acknowledge inspiration from many colleagues, despite naming only few:

I would like to thank to my supervisor Dr. David Obdržálek for providing me the opportunity to explore beauties of robotics in the past years. I truly appreciate his friendly, encouraging and empathetic approach.

Incidentally, the start of the COVID-19 pandemic helped me to meet Fernando Baltasar dos Santos Pedrosa at CERN, he lead a CERN against COVID taskforce focusing on prototyping a washable 3D printed face mask that I joined. Discussions with Fernando helped me in pinpointing the topic of this work, as he introduced me to the Robotics team in CERN BE-CEM group.

I would like to thank to Dr. Mario Di Castro, the Section Leader of the Mechatronics, Robotics and Operations Section in CERN BE-CEM group, to giving me the opportunity to explore sensing capabilities of a prototype of the Fraunhofer FMCW 80 GHz radar, with the only prototype CERN-wide!

Many thanks go to Dr. Roberto Cittadini, my colleague & partner in crime, without his help the sensing measurement sessions would have been much, much longer! I truly enjoyed our interactions, learning about biomechanics and computer vision applications from him!

Last but not least, many thanks belong to my dear parents, my family and my friends, who have tirelessly supported me throughout the the years of my education & professional life.

Contents

List of Figures

List of Tables

List of Abbreviations

AFE Analog Front End

CERN Conseil Européen pour la Recherche Nucléaire/European Organization for Nuclear Research

CFAR Constant False Alarm Rate

CW Continuous-Wave

CWT Continuous Wavelet Transform

FFT fast Fourrier transform

FMCW Frequency-Modulated Continuous-Wave

GDPR General Data Protection Regulation

IF instantaneous frequency

LHC Large Hadron Collider

LTS long-term support

SFCW Stepped-Frequency Continuous-Wave

RADAR RAdio Detection And Ranging

RR respiratory rate

UWB Ultra-Wideband

Structure of this document

This work studies an automated remote sensoric detection of human vital signs, with a Frequency-Modulated Continuous-Wave (FMCW) radar. The main objective is to assist a remote Operator in understanding whether a person, located somewhere remote in an industrial settings, breaths, and how well the person does so. We aim at remote sensoric detection of human vital signs, particularly respiratory rate, with a FMCW radar, and use two other sensors for a calibration of the radar measurement: a contact-based chest belt, and a contact-less distance laser.

The structure of the document is following:

In Chapter 1 we describe the scenery of such a respiratory activity detection, describe principles of radars, and highlight relevant contact-based and remote detection methods.

In Chapter 2 we discuss privacy and ethical aspects of this work.

In Chapter 3 we propose how to approach the outlined challenge to achieve the objectives, with the FMCW radar and two calibration sensors: from practical ethical aspects of the work, through description of the strain-sensing method of chest wall displacement that we use as one of the calibration measurements, through practical aspects of data acquisition, discussing why our three-sensor system does not have to be considered to be a distributed system, a brief overview of signal processing approach, to a list of possible scenarios for our multi-sensor measurement.

In Chapter 4 we summarize learning from the measurements with the FMCW Fraunhofer 80 GHz radar, and in Chapter 5 we detail on radar signal analysis approaches and results: we describe an application used for data acquisition, troubleshooting of radar data acquisition and signal processing, we continue with example of the signal processing cut flow, commence with description of an application we developed to study limitations of the parametric model to describe the respiratory rate detected from the radar signal, concluding with the overview of results of the respiratory rate detection from the radar data.

In Chapters 6 and 7 we describe measurements and signal processing leading to detection of respiratory rate, with the calibration sensors, the PLUX chest belt, and the Baumer OM70 distance laser.

In Chapter 8 we discuss the achieved results. In Chapter 9 we discuss possible directions to take to further expand on this work which are, however, beyond the scope of this work.

In Chapter 10 we conclude the outcomes of this work.

1. Introduction

In this work we explore remote sensoric detection of human vital signs with a FMCW radar. It originates as a R&D study with possible impact on Conseil Européen pour la Recherche Nucléaire/European Organization for Nuclear Research (CERN) Large Hadron Collider (LHC) Operations: we aim to automatically detect a human respiratory detection in an industrial settings, e.g. the one in the CERN LHC tunnel, with the radar sensor attached to a remotely-operated mobile robot. We aim to detect the human respiratory rate with a FMCW radar, and calibrate the radar detection with two other sensors: a contact-full chest belt, and a contact-less distance laser.

This chapter aims to introduce the reader to various aspects of the remote detection of human vital signs – in particular the respiratory rate – with a FMCW radar.

In Section 1.1 we set the scene and describe the objectives of this work. Then, Section 1.2 focuses on a brief description of principles of radars. Section 1.3 provides an overview of various aspects of the remote detection of human vital signs, that has been leveraged for the detection with a FMCW radar described in subsequent chapters.

1.1 Setting the scene

Imagine an environment with industrial settings, e.g. the one shown in Fig. 1.1, with technicians, engineers and scientists working in spaces that are a controlled environment, with a variety of possible safety challenges in terms of confined spaces, ambient temperature and wide temperature gradients, noise, strong (electro-)magnetic fields, high voltage electric circuits, atmosphere composition, water vapour and other gas leaks, heavy machinery and equipment, fire hazard, etc. where people work together in peace to push frontiers in fundamental science research. Such is the environment one can encounter at CERN.

CERN keeps health and safety of its personnel as the highest priority, provides safety training and governing procedures. The CERN Fire and Rescue Service respond to any safety incidents on-site. As the CERN campus spans over several sites around the LHC (the circumference of the LHC tunnel has 26,659 m), it can take up to 22 minutes to reach the site of an incident.

The Organization explores ways to benefit from automating tasks which can be automated, and leverages various robots with for sensoric monitoring of the infrastructure, equipment, environment, and possibly the health of the personnel.

During the works in the challenging industrial settings environment outlined above, members of personnel may trip and fall, e.g. as shown in Fig. 1.2, and without checking their consciousness status and/or vitals, their colleagues (or Fire & Rescue operators) do not know whether the fall was due to inattention when one was thinking deeply and overlooked an obstacle, or because the person lost consciousness. Either way, checking up on the person in need is in order, and that may require checking their vitals.

During a cardiac arrest event every second counts, it is paramount to assess the health status of the person as soon as possible and start providing measures

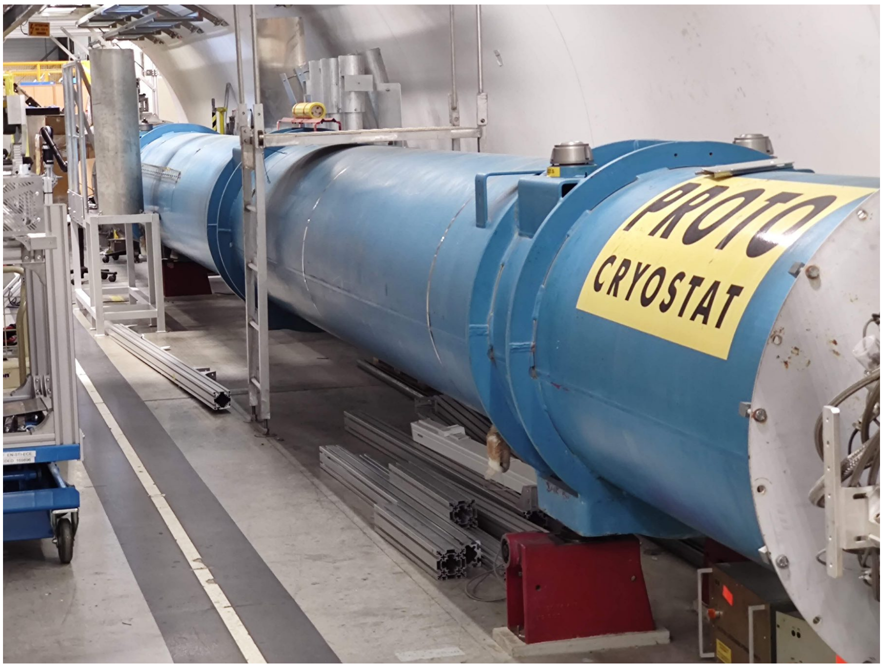


Figure 1.1: Simulation room to replicate the LHC tunnel environment, used for development of robotic systems and for testing their functionality.

of first aid, whether as a bystander, or as a first responder professional. Having means of remote health monitoring can help delivering help faster. A mobile robot equipped with a sensor that can assist with a remote sensoric detection of human vital signs is one of the means, and it is where this work comes into play.

Lately, many wearable/smart gadgets that can serve as personal health monitors are available, but, unfortunately, are not suitable for the environment at CERN, for reasons discussed further in Chapter 2. However, a mobile robot equipped with sensors suits better to CERN needs.

There are dozens of mobile robots in the LHC tunnel for monitoring purposes (of infrastructure and equipment), therefore equipping them or additional mobile robots with a suitable sensor or a sensor set may bring benefits in terms of remotely monitoring also vital signs of personnel in need, should the situation require that. With a suitable equipment, in such a situation, a Fire & Rescue operator would be able to remotely check and assess vital signs of a person, notably the person's respiratory rate, and communicate with other personnel on the incident site, or with the dispatched Fire & Rescue unit.

Such a "mobile robotic probe" can be equipped with a variety of sensors, not only a FMCW radar, but also a IR camera to visualise temperature gradients in the environment and their movement in time, a microphone for a possible two-way communication.

Using a radar for health/vital signs monitoring is not a new concept, although a different types (UWB radar, further discussed in Section 1.3) and in well defined, static settings: a radar is fixed to a ceiling of a hospital room, directed downwards to a hospital bed where a person is lying down (dressed in a thin hospital gown), and their vital signs are remotely monitored.

Having a FMCW radar attached to a mobile robot and performing remote detection of human vital signs in a very dynamic environment brings a completely

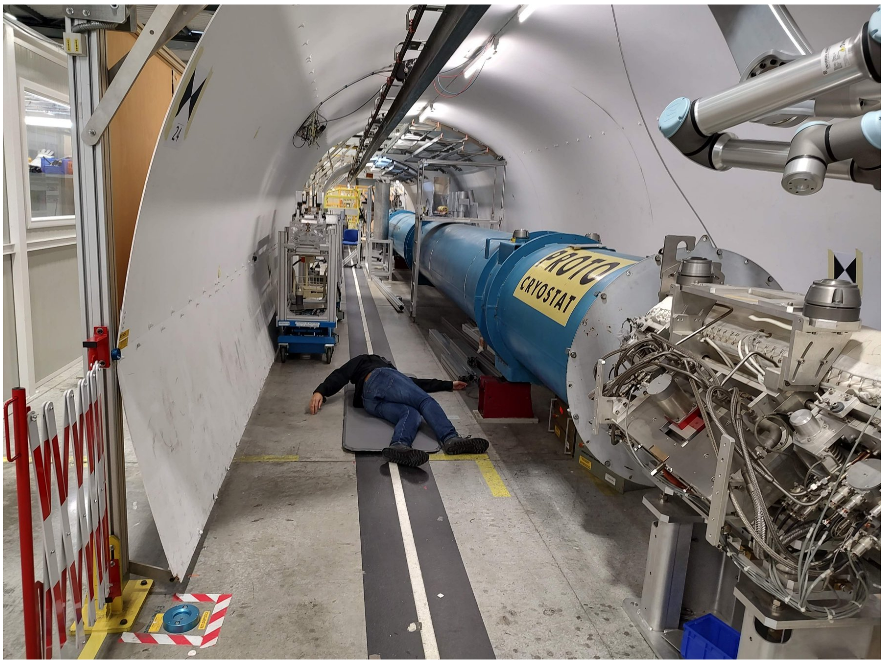


Figure 1.2: Simulation: A person fell in the tunnel, and now is laying down on their side, with their legs directed towards the observer. The observer may be e.g. a mobile robotic system equipped with a FMCW radar sensor.

different set of challenges: from aiming/targeting and detecting the human's position, through screening/hiding the respiratory activity through different types of clothing, to detecting the respiratory rate from signal acquired by the radar and its subsequent signal processing, that comprises from applying the suitable model parameters to the signal processing (as we discuss throughout this work).

In this work we explore the limitations of a FMCW radar-equipped mobile robot, for a remote detection of human vital signs, particularly the respiratory rate, in an industrial settings environment. We are interested in parameters that lead to successful signal gain and processing, and also in limitation of signal acquisition and processing. We are interested in a real-world environment, in understanding the realistic limitations in terms of human's body position, breathing patterns, human's clothing limitations, environmental conditions including obstacles in the line of beam. The objective of this work is to understand the limitations, to understand what to expect and manage our expectations, but not bring up a product from a conceptual model to a complete production-ready final product.

Let's not get ahead of events – please accept our invitation to embark on a journey of learning and exploring how technology can help humanity discover the beauties of the Universe with the help of remote sensory robotics!

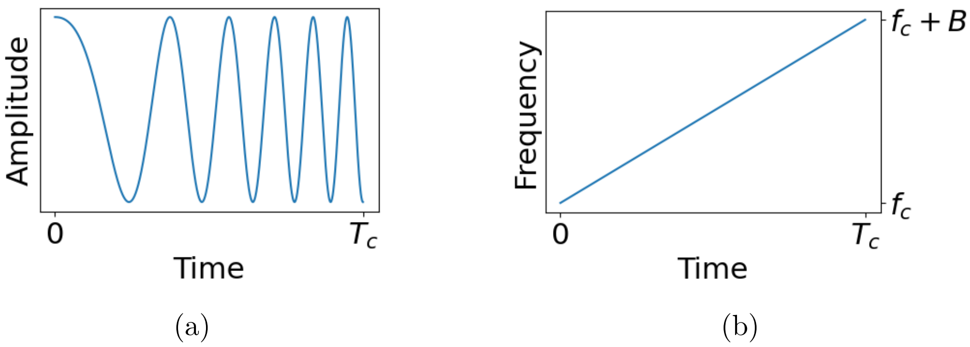
1.2 Radars

A RAdio Detection And Ranging (RADAR) [1] is a device used for radio-location with radio waves to estimate a distance (ranging), angle (azimuth), and radial velocity of objects with respect to the radar location.

1.2.1 A FMCW radar fundamentals

Now introducing several terms to describe fundamentals [2] of a FMCW radar.

Chirp is a signal that a FMCW radar transmits. It is a frequency-modulated sinusoid, where frequency increases linearly in time. Fig. 1.3 shows an amplitude-time and frequency-time plots of a chirp. A chirp characteristics are the start frequency f_c, the bandwidth B, and the duration T_c.



(a) (b)

Figure 1.3: A representation of a FMCW radar chirp. a) the amplitude-time plot, b) the frequency-time plot of a chirp.

Radar schema is depicted in Fig.1.4: a *synthesizer* generates a chirp, that is transmitted by a *Tx antenna* and also sent to a *mixer*. The chirp is reflected from an object and the reflected chirp is received by a *Rx antenna* and sent also to a mixer. Mixer takes two inputs, from the Tx and Rx antennas, mixes them and produces an instantaneous frequency (IF) signal that is analyzed further on.

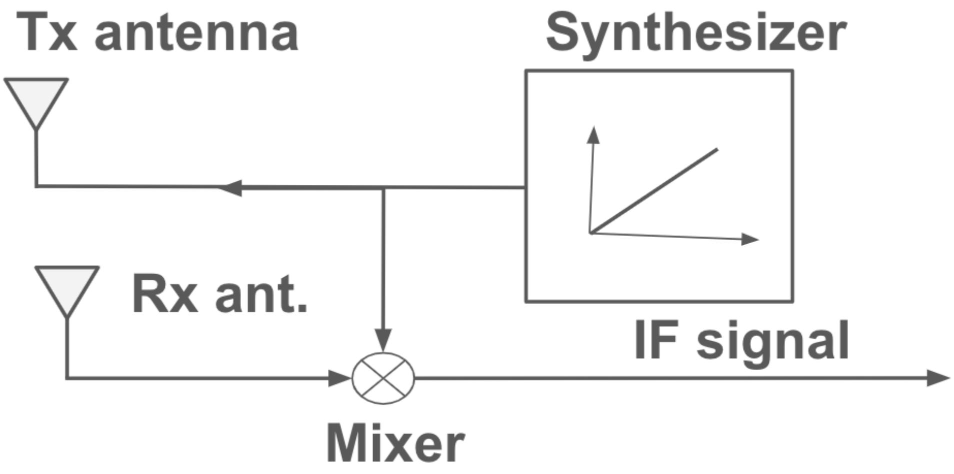


Figure 1.4: A schema of a radar.

Mixer is a device that mixes two input signals (Tx, Rx) into one output signal (IF). Let the two input signals be

$$x_{\mathrm{Tx}} = \sin(\omega_1 t + \phi_1), \tag{1.1}$$

$$x_{\mathrm{Rx}} = \sin(\omega_2 t + \phi_2). \tag{1.2}$$

The output IF signal represents difference of IF frequencies of the two input signals:

$$x_{\mathrm{IF}} = \sin[(\omega_1 - \omega_2)t + (\phi_1 - \phi_2)]. \tag{1.3}$$

Fig.1.5 shows frequency-time plots of the inputs, the Tx- and Rx-chirps, and the resulting mixing output, the IF signal. The plot a) shows the Tx-chirp and the reflected Rx-chirp. In case of a single object, the Rx-chirp is only a time-delayed version of the Tx-chirp. Following (1.3), the resulting IF signal is a constant frequency, as shown in plot b). Let S be the slope of the Tx-/Rx-chirps in the frequency-time plot, and τ the time of round-trip between the radar and the object. The frequency of the IF signal is then

$$S\tau = 2Sd/c, \tag{1.4}$$

where d is the distance between the radar and the object, and c is the speed of light in the environment (usually in the air).

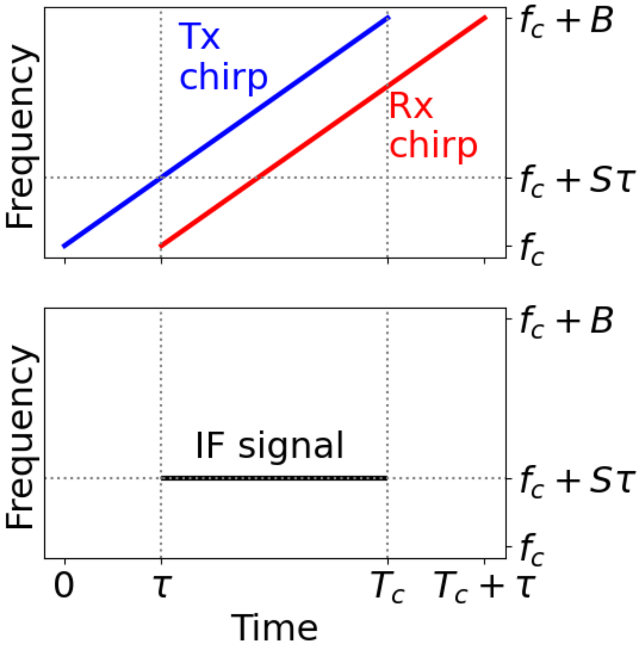


Figure 1.5: Mixing of Tx and Rx signal into a IF signal. Top: the frequency-time plot of the Tx-chirp and Rx-chirp; Bottom: the frequency-time plot of a IF signal output.

IF signal analysis employs Fourier Transform, in particular fast Fourrier transform (FFT), converting a time domain signal into the frequency domain. A sinusoid in the time domain transforms into a peak in the frequency domain, as shown in Fig. 1.6.

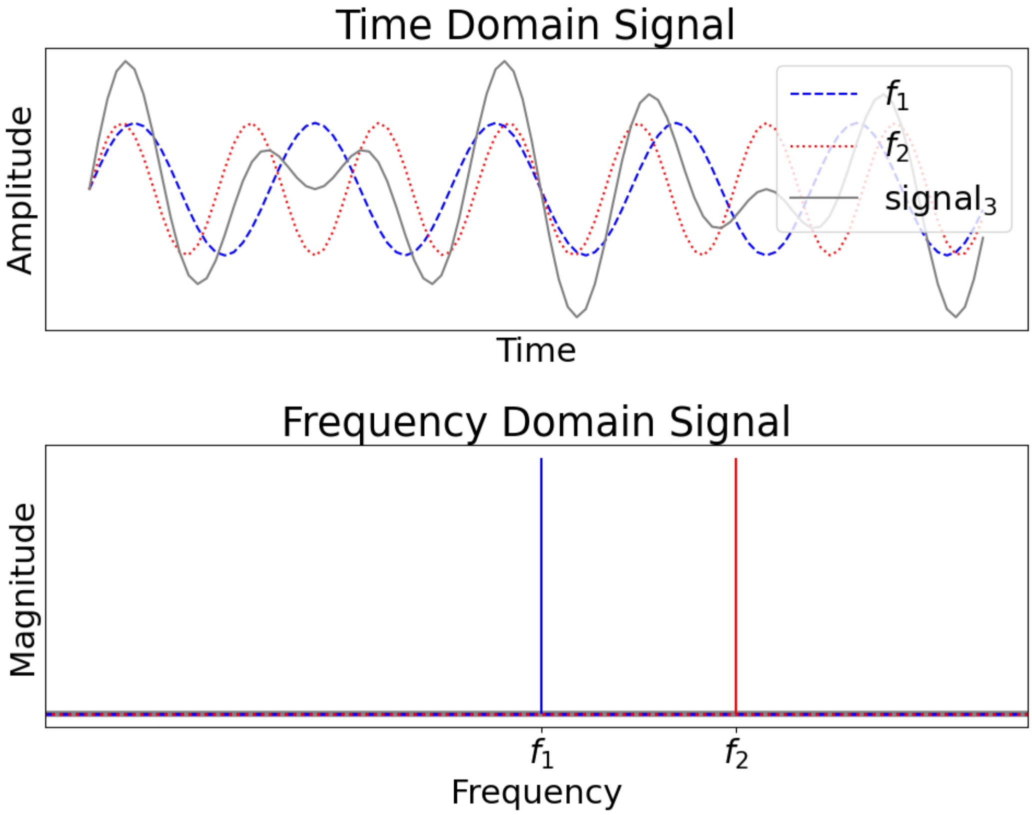


Figure 1.6: A Fourier transform of a sinusoid in the time domain (top) into a peak/spike in the frequency domain (bottom). The top plot shows three signals: The first signal (blue) corresponds to a sinusoid with frequency f_1, the second signal (red) corresponds to a sinusoid with frequency f_2, and the third signal (gray) is composed from the first two signals. A FFT applied converts time domain signal to frequency domain. The first signal (blue) then shows a spike at frequency f_1, the second signal (red) shows spike at frequency f_2, and the composed third signal (gray) shows spikes at both frequencies $\{f_1, f_2\}$ of its both components.

A IF signal of a single object at a distance d from the radar is

$$x_{\mathrm{IF}} = A \sin(2\pi f_{\mathrm{IF}} t + \phi_{\mathrm{IF}}), \tag{1.5}$$

where

$$f_{\mathrm{IF}} = 2Sd/c, \tag{1.6}$$
$$\phi_{\mathrm{IF}} = 4\pi\Delta d/\lambda, \tag{1.7}$$

with λ being the wavelength of the chirp. As the phase of the IF signal is quite sensitive to small changes in the range of the object, using radars for ranging and hence, detecting movements of chest wall, is a suitable approach to remotely detect vital signs: radar signal processing will help detect small chest wall displacement changes and will highlight them with respect to other possible random movements registered in the environment.

1.2.2 Different types of radars

Based on the type of signal transmitted, four types [3] of radars are recognised:

- Continuous-Wave (CW) radars,
- pulsed, Ultra-Wideband (UWB) radars,
- Stepped-Frequency Continuous-Wave (SFCW) radars, and
- Frequency-Modulated Continuous-Wave (FMCW) radars.

Their features comparison is in Table 1.1. This work describes measurement with a FMCW Fraunhofer 80 GHz radar (further described in subsection 1.3.3). Despite a FMCW radar might be used for multi-object detection and range estimation, it is essential to know a general area where the subject is, and for a vital signs detection to target the area around diaphragm, as we discuss later in Chapter 4. A study of breathing movements [4] is a great resource for understanding the ability to detect respiratory rate.

Remote monitoring of respiratory rate

Method	Multi-object detection	Range estimation	Power consumption
FMCW	Yes	Yes	High
CW	No	No	Medium
SFCW	Yes	Yes	Medium
UWB	Yes	Yes	Low

Table 1.1: Comparison of methods of remote monitoring of respiratory rate.

1.3 Vital signs detection

This section outlines methods and tools for contact and remote vital signs detection, notably for respiratory rate (subsections 1.3.1 and 1.3.2). Subsection 1.3.3 describes sensors used for respiratory rate detection and measurement calibration, and the data acquisition setup for these sensors in subsection 3.2.

1.3.1 Contact-based detection of respiratory rate

Discussing two groups of contact detection methods [3] for respiratory rate, compared in in Table 1.2: methods sensing perception of breathing activity of the human body (e.g. displacement of the chest wall), and methods comparing properties of inhaled and exhaled air (e.g. air composition, temperature, humidity).

A contact detection with piezo-electric resistive sensor was used to detect respiratory rate through chest wall displacement, as an independent calibration method to the remote detection of respiratory rate with the FMCW radar. The sensor is further described in subsection 1.3.3.

Contact-based respiratory rate monitoring

Mechanical displacement sensing of chest			
Method	Accuracy	Long-term monitoring	Caveats
Strain-based	High	Yes	Probe tightly attached
Impedance pneumography	High	Yes	Motion artefact effect
3D movement sensing	Medium	Yes	Cost

Air based sensing			
Method	Accuracy	Long-term monitoring	Caveats
Air component	High	No	Environment
Air humidity	High	No	effects
Air temperature	High	No	–

Table 1.2: Comparison of methods of contact-based respiratory rate monitoring.

Chest wall displacement sensing method measures extension of the human subject's chest while breathing: during respiration the diaphragm contracts and expands together with the rib cage and muscles, allowing the air enter and leave the lungs. The activity of the diaphragm and respiratory muscles leads to displacement of the chest, which expands up to 7.37 cm on the circumference [5]. Further details are provided in Secion 3.1.

Air composition-based method focuses on measuring level of CO_2 in the air that was inhaled (approximately 0.04 %) or exhaled (around 6 %, [6]). The difference in CO_2 levels can be measured using chemical sensors (e.g. infrared or fiber optics) to determine the respiratory rate, through the capnography technique. There are two main capnography setups: the mainstream and the side stream method. In the side stream approach measurement, the sensor and the main processing units are placed away from the human. On the contrary, for the main stream approach measurement, the sensor is located between the processing unit and the sampling tube, which is attached to the facial area of the human with a face mask. The main stream approach is faster and more accurate, however is more expensive and can be inconvenient to the human subject, as the senor heats up about 40 °C. The measurement of volume of expired CO_2 over time resembles a wave / sinusoidal shape, called a capnogram. The capnography measurement is not comfortable for long-term monitoring, and can be sensitive to other air components and also to ambient changes in humidity and temperature [7].

In this work we are targeting remote detection of respiratory rate as a non-intrusive mean in general, therefore the face masks are not suitable for this approach as they are somewhat limiting person's comfort. However, a CO_2 detector may be attached to a robotic arm and "sniff" the environment in this way.

Air temperature-based method concentrates on measuring the temperature difference of the inhaled and exhaled air. The difference can be as high as 15 °C, while for the ambient temperature of $T_a = 23$ °C it is rather within a 4 °C difference range [8], depending on the air humidity. The measurement is performed with a temperature sensor attached to a face mask, temperature measurement is converted to an electric signal (voltage or current difference), which is enhanced by an analog interface. Then the signal is processed to provide a respiratory rate. Possible sensors for this type of measurement are thermistors, thermocouples, pyroelectric sensors or fiber optics sensors.

Air humidity-based method is similar to capnography method, determines the respiratory rate through measurement of water vapor level of the inhaled and exhaled air. The humidity of inhaled and exhaled air differs by approximately 20 % to 60 % [9]. The experimental setup for an air humidity-based method is similar to that of air temperature-based measurement, while the sensor in question is a humidity sensor of capacitive or resistive type.

1.3.2 Remote detection of respiratory rate

Remote detection of respiratory rate (and heart rate) can be performed with a help of radars. Detection of vital signs with radars depends on the effect of radio signal modulation. Schema of the vital signs detection is in Fig. 1.7. Radar emits a signal through its transceiver, the signal travels through the environment to the human subject, where it reflects. While a living human breaths, their chest wall expands and contracts, which causes the radar signal modulation. The reflected signal contains information about human's respiratory and heart rate, together with environmental and electronic noises. The signal is received through the radar's receiver. During the signal processing, removal of the noise occurs – more details in section 3.3 – and clean human's vital signs, the respiratory and heart rates, are detected. Typically, the the power transmitted by the radar within a two-meters distance does not exceed 12 dBm (less than that of a smartphone), therefore these radars are safe. A more detailed discussion of four types of radars for vital sign monitoring is in Section 1.2.

1.3.3 Sensors

A remote sensoric detection of respiratory rate in industrial settings with a FMCW Fraunhofer 80 GHz radar is studied. In order to understand the limitations of the setup, a series of calibration measurements has been performed, where the radar respiratory detection has been augmented with a PLUX chest belt measuring chest wall expansion, a Baumer laser device measuring distance, and also by counting respirations. These simultaneous measurements help in assessing how confident one can be with the data analysis package developed to process the radar signal and detect respiratory rate.

FMCW Fraunhofer 80 GHz radar. The aim is to detect human respiratory rate with a FMCW Fraunhofer 80 GHz radar [10], [11]. In Fig. 1.8 the radar is depicted, a small bulb-like object, with less than 5 cm in any of the directions.

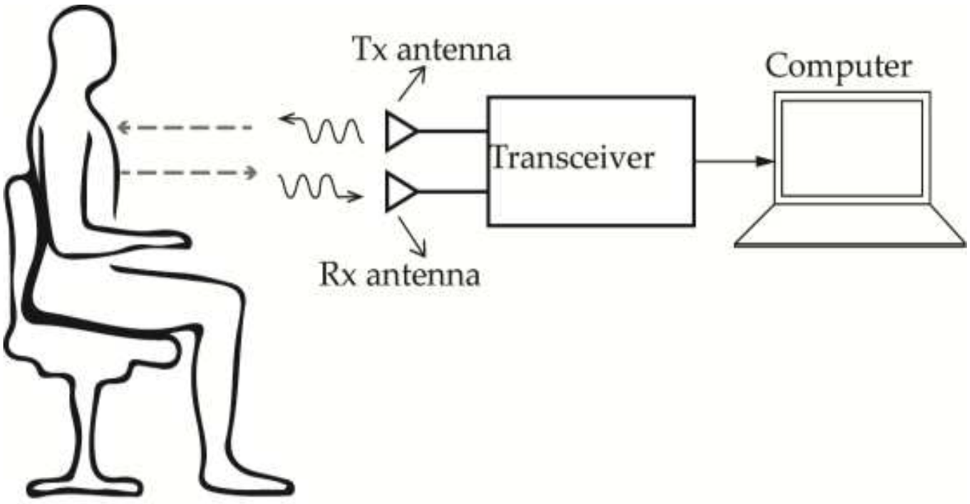


Figure 1.7: Schema of a remote detection of respiratory rate with a radar, as shown in Fig. 9 of [3]: a human subject's respiratory activity is registered by a FMCW radar, and recorded & processed in a computer.

The radar consists of two boards: a high-frequency front-end and an acquisition back-end, that includes intermediate frequency pre-amplifier, an anti-aliasing low pass filter and a 16-bit analog to digital converter with a sample rate of 1 MSa/s. The radar can be connected to a computer through a standard USB cable, which serves as both the power source and data link for the radar.

Bamuer OM70 distance laser. As one of the calibration measurements, a Bamuer OM70 distance laser [12], shown in Fig. 1.9, has been used to measure chest wall displacement of the human subject, and detect the respiratory activity remotely also in this way. The laser device of class 1 emits light at $\lambda = 660$ nm, and the device measures distance from a range of 150 mm to 1400 mm. A 3D printed mount was used to fix the FMCW Fraunhofer 80 GHz radar and the Bamuer OM70 distance laser together, to leverage the distance laser as a laser pointer and a distance measure simultaneously, as well to be sure about the direction of the radar's peak beam.

PLUX chest belt. As another one of the calibration measurement, this time a contact-based detection, a PLUX chest belt [13] has been used. It is shown in Fig. 1.10. It is a piezo-electric strain-based sensor of chest wall displacement. It was affixed the chest belt tightly to the human subject's chest and ensured the box with power source and bluetooth communication dongle do not limit human's comfort, that would effect their respiration.

Other possible sensors that could complement the vital signs measurement with a FMCW radar, however are beyond the scope of this work and have not been included in the measurements, are additional sensors for the vital signs detection combination, e.g.

Figure 1.8: FMCW Fraunhofer 80 GHz radar used to study remote sensoric detection of respiratory rate.

- a microphone: since 2020, NASA's Perseverance Rover is equipped with two microphones [14], and they are successfully capturing sounds from the activity of the Rover and other robotic systems on Mars [15], [16]. Therefore, we may consider addition of such an additional sensor to a robotic system carrying the FMCW radar to remotely detect human vital signs.
- a thermal infra-red camera [17] for thermal imaging: such a sensor can provide additional information about the human subject and the ambient environment, and possibly also about movements of the human subject, which can be a beneficial source of information in case the detection of vital signs does not succeed with the FMCW radar – the nature of human subject's clothing influences significantly the respiratory rate detection by a FMCW radar, therefore it is paramount to have other alternative means of detection at (a robotic) hand.

Figure 1.9: Bamuer OM70 distance laser used to study remote sensoric detection of respiratory rate.

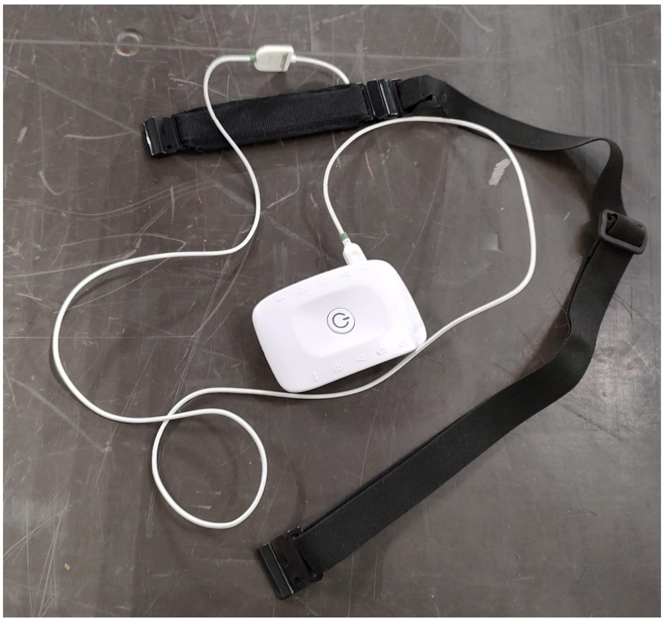


Figure 1.10: PLUX chest belt used to study remote sensoric detection of respiratory rate.

2. Privacy & ethics

This chapter aims to bring up various privacy and ethical aspects of the remote detection of human vital signs.

Section 2.1 outlines privacy consideration for the remote detection of the human vital signs, detailing why a remote robotic sensor is more suitable than a smart wearable gadget. Section 2.2 describes ethics considerations of an activity that handles such a vital signs detection. Further practical aspects are discussed in Section 2.3.

2.1 Privacy considerations

For over a better part of the past decade, there are many wearable/smart devices on the market that can measure human vital signs in a contact manner, such as smart watches or heart belts. These devices are fairly affordable (costing from several tens Euros), one can monitor their vital signs (e. g. heart beat) with them and fetch the information through a proprietary mobile application, export less detailed data to perform data analytics, or share their activity performance report with others.

One disadvantage of these smart devices deployed in an industrial settings is privacy and protection of data. The vital signs data is collected by a proprietary hardware, passed to a 3rd party (usually the manufacturer of the device) in a detailed format, often including personal data about the vital signs, location of the person, and are attached to a user account with another set of personal information available.

Processing such an amount of personally identifiable information is not a straightforward task, in terms of data privacy and protection, data processing, information security, infrastructure, and legal point of view. It can often become a nightmare that can result in high cost for the organization, outweighing the possible benefits of monitoring health of the personnel in places that are difficult to reach on a short notice.

Although in the majority of Europe, data protection is regulated by General Data Protection Regulation (GDPR) [18], that is not the case in the international organizations such as CERN. Nevertheless, data privacy and protection at CERN is regulated by an Operational Circular [19], that lists duties and responsibilities of a service manager in that area.

Wearing a smart device is a personal decision of each person, and it would be quite difficult to mandate it and ensure safe and secure operation at scale to monitor vital signs of all personnel. As discussed earlier, at the same time, being able to spot a health emergency and dispatch a unit of first responders to a location that is difficult to reach, while providing monitoring of vital signs remotely, is a very useful capability, and therefore, studying potential and limitations of a remote detection of human vital signs with help of a FMCW radar connected to a mobile robot is the next best approach, viable in the legal framework of CERN.

2.2 Ethics considerations

2.2.1 Objectives

The objective of this work is to explore and understand limitations of a FMCW radar for a remote detection of human vital signs, namely respiratory rate (and as a stretch goal possibly also heart rate), in industrial settings.

2.2.2 Environment

The R&D phase of the project is carried out in a way gentle to the environment. An already assembled/existing radar sensor and other computer equipment has been used, and a laboratory space leveraged to conduct the measurement. The utilities consumption overhead is minimal, and consists of power consumption of the radar sensor. We do not make use of any liquids or fossil fuels to directly conduct the research, therefore we do not introduce any foreign agent into the local environment. Local transportation of the equipment across the laboratory is performed in an environment-friendly way (walking).

2.2.3 Stigmatisation & Exclusion, Malicious Exploitation

This research is inclusive, strongly supporting diversity of the CERN's Member of Personnel as one of the CERN's core values. In fact, the core research team represents women and men equally. The R&D phase of the project excludes a malicious exploitation of the project, similarly the possible future production phase will exclude it, based on CERN's internal regulations.

2.2.4 Ethics Issues

This research project involves human participants, and involves non-invasive physical intervention of human participants. The research project involves personal data collection and processing, including further processing of the collected personal data in order to optimise the data analysis algorithms. This research project does not involve human embryos / fetuses, human cells / tissues, nor animals.

2.2.5 Data Privacy Protection and Security

The personal data collection and processing is carried out in a way compliant with the CERN's Operational Circulars, particularly Operational Circular No. 11 [19] and Operational Circular No. 5 [20]. The data is stored on facilities of the Organization. Access to data is granted only to the core research team members, in order to develop and optimise a SW package for data analysis.

2.2.6 Safety

The R&D phase of the project is carried out with safety in mind in every step of the project. All members of the core research team have been trained in the basics of safety in the workplace. Additionally, a member of the core research team has passed the CERN First Aider certification.

The laboratory environment (place where the measurement takes place) is secured prior to commencing the measurements, and adapted as needed during the experimentation sets, to assure safety of the personnel and equipment at all times.

2.2.7 Risk / Benefit Assessment

The research involves minimal risk to participants:

- physical risk (e.g. pain, bruising and infection, muscle soreness and pain as a consequence of exercise testing, health emergencies),
- psychological risk (e.g. stress associated with experiments and testing),
- social risk (e.g. invasion of privacy, loss of community standing),
- legal risk (e.g. criminal prosecution), and
- economic risk (e.g. loss of employment, loss of potential monetary gain).

The research provides no prospect of direct benefits to individual participants, but likely will yield generalizable knowledge to further society's understanding of this remote detection technique.

2.3 Practical ethical aspects

2.3.1 Methodology

In the R&D setting employed in this work's project, the radar is placed on a solid surface, detecting human vital signs of up to two human test participants (the core research team), in order to understand limitations of the detection (human respiration observation, device capabilities, environment) and to develop a SW package to perform the data analysis, possibly providing and comparing several approaches to data processing.

In a possible future production setting, the FMCW radar is meant to be mounted on an arm of a mobile robot. The robot would be controlled by a remote operator, a member of a Fire & rescue team. The robot equipped with a FMCW radar sensor would remotely monitor vital signs of personnel in an emergency situation, e.g. in an accelerator tunnel.

2.3.2 Legislature & Jurisdiction

This research project is carried out at CERN, therefore it is governed by CERN's Operational and Administrative Circulars and regulations.

2.3.3 Research Impacts

The impact of the R&D phase of the project results in understanding of limitations of the FMCW radar w.r.t. the conditions of the industrial settings. The following characteristics are considered to contribute to the “ambient environment” limitations:

- distance of the sensor from the human participant,
- direction focus of the sensor (including angular dependence of the radar beam cone on the signal gain),
- obstacles around the human participant (notably along the line of sight),
- changes in the local atmosphere composition (e.g. smoke or dust dispersed in the atmosphere),
- dependence of the sensor temperature on its operationality (and hence, a need for an active cooling mechanism).

Additionally, dependence of the human vital sign detection on following conditions is studied:

- clothing characteristics of the human participant (e.g. different upper body clothing layers, such as a t-shirt, sweater, jacket),
- human participant body position (e.g. sitting / lying down, detection from different parts of the body),
- human participant breathing patterns (e.g. standard breathing, fast or slow breathing, shallow or deep inhalation, gasping).

Understanding these conditions and their impact is fundamental for development of the SW package for data analysis, and interpretation of the detected human vital signs.

3. Solution proposal

This chapter proposes the approach to address objective of this work, i.e. to detect human vital signs, namely respiratory rate, with a FMCW radar.

Practical aspects of ethical consideration of this work have already been described in Section 2.3. Here in Sections 3.1 and 3.2 we describe data acquisition techniques. Section 3.3 outlines the general data analysis flow for the measurements with the three sensors, and later describe six different measurement scenarios in Section 3.4.

3.1 Strain-sensing of chest wall displacement

The most common methods of the chest wall displacement measurement are based on strain sensing, transthoracic impedance sensing or impedance pneumography and movement sensing using accelerometers, gyroscopes and magnetometers. The strain-sensing method uses resistive, capacitive, inductive and fiber optic sensors to measure the instantaneous change in strain. Piezo-resistive strain sensors, referred to as ”strain gauge” [21], change their shape when a physical displacement is applied. The change of shape causes a change of their resistance values, measured with an electronic circuit depicted in Figure 3.1. The piezo-electric sensor is embedded in a textile chest belt, attached to a human subject chest. The sensor signal is read out to a analog interface for enhancement, then processed to measure the respiratory rate.

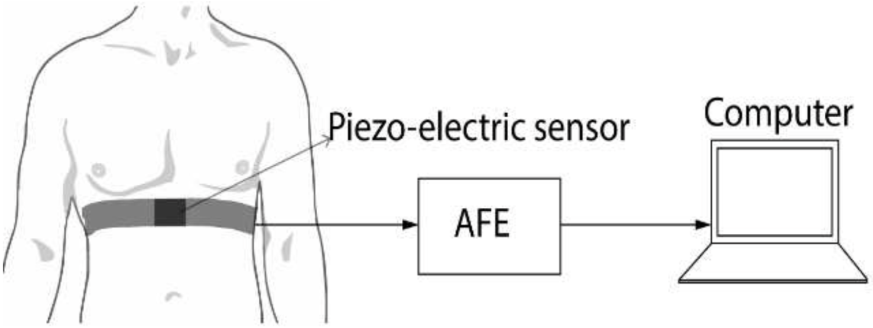


Figure 3.1: Electronic circuit utilized in a measurement of a chest wall displacement with a piezo-electric sensor, as shown in Fig. 5 of [3]: the signal from piezo-electric sensor (part of the chest belt) is read out through an Analog Front End (AFE), and then processed in a computer.

3.2 Data acquisition

The data will be acquired from the 3 sensors independently on each other, with emphasis on time interval overlap in measurement in this distributed system. The series of measurements will be performed in a controlled laboratory environment with ambient temperature 20-23 °C.

FMCW Fraunhofer 80 GHz radar data acquisition. For a remote detection of respiratory rate, the radar will be mounted to a pad in the vicinity of a data collection computer. Radar will be connected to the data collection computer via a USB cable, providing power supply and data channel simultaneously. Without additional active cooling, the radar may overheat within 30-45 minutes of being plugged in the USB port, resulting in inability to temporarily further acquire data. In such a situation, the radar will be unplugged for 10-15 minutes to cool it down in an ambient room temperature, and the measurements will resume later. The radar data acquisition package is written in C++, based on CERN Robotics Framework [22], leverages serial port communication for data acquisition.

Bamuer OM70 distance laser data acquisition. For data acquisition, the distance laser will be mounted to a pad with the radar, allowing for simultaneous distance measurement directing to desired location. Laser will be connected to the same data collection computer through a FTD 232 device USB cable. Although the different models of the laser may be able to measure distances in a variety of distance ranges, the one particular used allows to activate and leverage only a single range of 0.15 m to 1.5 m. This distance will be sufficient for simultaneous chest-distance and radar measurement, for larger distances only the laser pointer functionality of the laser sensor will be used to direct the radar appropriately, losing the distance measurement capability. The laser data acquisition package is written in Python 3, benefiting from the PySerial [23] library.

PLUX chest belt data acquisition. The chest belt can connect to a data collection computer via the Bluetooth interface. While the data acquisition from the radar and the distance laser has been achieved on a computer running a LTS flavor of Ubuntu Linux OS, attempts to connect the chest belt device on the same box were not successful, therefore a Windows OS box will be used for the chest belt data acquisition. The chest belt data acquisition package is written in Python 3.

3.2.1 Synchronization of data sources

This work focuses on remote detection of respiratory rate with a FMCW radar. For calibration of radar measurement (and radar signal processing) we will use two other different sensors, and manual count of respirations per minute. Each measurement will last 1 minute, the three sensors will be part of a very loosely coupled distributed system. The radar and the laser will be connected to one computer, and the chest belt to another computer. Although the sensors and their computers constitute a distributed system, we do not need to apply complex distributed systems algorithms to assure time synchronization in the system: the important feature of the system is that the measurement period of 1 minute is longer than the time of 1 respiration (a typical human respires between 10 and 16 times per minute), and very precise synchronization among the ”nodes” is not needed: we need to match 1 minute interval with few seconds precision. Additionally, we do not need a monolith program to acquire and process data from each sensor, and it is sufficient to have data acquisition and processing for each sensor separately.

3.3 Signal processing: from measurement to detection of respiratory rate

The FMCW and CW Doppler radars signal processing allows for respiratory rate and heart rate extraction through time and frequency domain analysis. Peak detection methods are used as a time-domain method to detect peaks in respiration (and heart) signals. The FFT and statistical methods in time-domain data help differentiate between time-varying and static signals [24].

The signal processing of a CW Doppler radar[1] has been described in literature, e.g. [25]. In short, data is acquired from the radar as a data frame, with 4096 channels per chirp, and time resolution of 50 ms, for period of 1 minute. The data frame is passed through FFT to move to time-domain and to be able to further detect peaks in the respiratory activity. The data is passed through a Kalman filter to estimate the peaks. Then the peak outliers are removed by calculating the variance of the radar data packet, and evaluating the outliers with respect to the packet mean feature. The signal processing continues by removing variance outliers, unwrap the phase of the processed signal, and concludes the search for a respiratory rate with a series of forward and inverse FFTs.

Several situations have been encountered when the respiratory rate detection from the radar signal was not successful, and attempted to adapt the signal processing algorithm parameters to recover the respiratory rate from signal:

- Peak outliers removal: When the outliers used to be evaluated with respect to the packet average feature, it was often impossible to detect respiratory rate from the measurement. Therefore a decision has been made to evaluate the outliers with respect to the packet mean feature.

- Shallow breathing or movement of the human subject during the data acquisition: the respiratory rate detection approach is fragile with respect to additional dynamics, and when either of the movement types is present, it often results in inability to detect respiratory rate from the processed signal, due to variance higher than a cut-off threshold that is imposed on the data. An artificial experimentation with increasing the cut-off threshold may be put in place, however, rapid movements may end up being the killer parameter of the whole detection.

3.4 Multi-sensor measurements

With the 2 remote and 1 contact sensors a series of measurements has been performed to understand the limitations of the setup, the environment, and also to restrict areas on the human subject's body that allow for a successful respiratory rate detection.

---

[1] Principles of a FMCW and a CW Doppler radar signal processing are same.

Measurements with the 3 sensors have been performed independently. Each radar measurement collected data for 1 minute, and the focus of data analysis was on time interval overlap of the 3 sensor measurements and the manual respiratory rate count. Further describing the scenarios of respiratory rate detection that was performed:

`M01`: 1D angular resolution spectrum. We would like to understand the 1D angular resolution spectrum of the radar, therefore we will perform a measurement to attempt to understand the radar behavior when the direction of the radar's center beam differs by multiples of 5 ° steps.

`M02`: Subject lying down on their back. We would like to understand how well / whether we can detect respiratory rate of the subject lying down on the back, pointing the FMCW radar at different parts of the subject's body, e.g. head, shoulders, several chest locations, and hips and thighs. The distance of the radar & laser from the subject will be fixed, in the range of 1.1 m to 1.5 m.

`M03`: Subject lying down on their side. We would like to understand how well we can detect respiratory rate of the subject lying down on their side, either facing the radar, or having the radar pointed at their back. While the subject will be facing the radar, we will focus the measurement at the subject's collar bone, several locations on their chest, waist and hips. While detecting from the person's back, the measurements will direct radar to the person's upper shoulders, scapulas, several places on the chest around their diaphragm, and waist and hips.

`M04`: Subject lying down on their stomach. We would like to understand how well we can detect respiratory rate of the subject lying down on their stomach, turned with their side to the radar were detected. The radar will be directed from the person's side to locations on their shoulders, scapulas, rib cage line, waistline, and hips.

`M05`: Subject sitting down, wearing a standard-issue work jacket. We would like to understand how well we can detect respiratory rate of the subject sitting down in a chair. The radar will be directed to the subject's diaphragm area.

`M06`: Subject sitting down, wearing a standard-issue work jacket, with a plexi-glass in the line of beam. We would like to understand how well we can detect respiratory rate of the subject sitting down in a chair similarly to M05, but with a clear plexi-glass obstacle to the line of beam.

4. Measurements with the radar

This Chapter describes several series of measurement that we performed in order to understand the constraints of remote sensoric detection of human vital signs, particularly the respiratory rate, with a FMCW radar.

Firstly, we show overview of respiratory rate (RR) detected by radar, counting, chest belt and distance laser, with baseline parameters of the detection model, in Section 4.1. Then we describe the initial get-to-know measurements in Section 4.2, performed only with the radar. In the subsequent Sections 4.3 through Section 4.8 we describe the measurements of scenarios `M01` to `M06` where we measured the RR with radar, and calibrated the measurement with other sensors in parallel.

Considering the privacy nature of the data acquired with the FMCW radar, the data cannot be included in the attachment folder.

4.1 Baseline detection overview

In Fig. 4.1 we present a baseline configuration of RR measurement with different sensors: the radar, chest belt and distance laser. In addition, for measurements with Measurement ID larger than `0035`, we counted the RR during the data acquisition. We will discuss the detection parameters with different sensors in Chapters 5 – 7. From Fig. 4.1 and tables in this Chapter one can see that the baseline detection method for the radar RR detection is not perfect: the Radar RR data points are far away from the Counted RR data points and outside the gray band, and that there are scenarios when we need to better understand the limitations, to deliver a better detection model, discussed in Chapter 5.

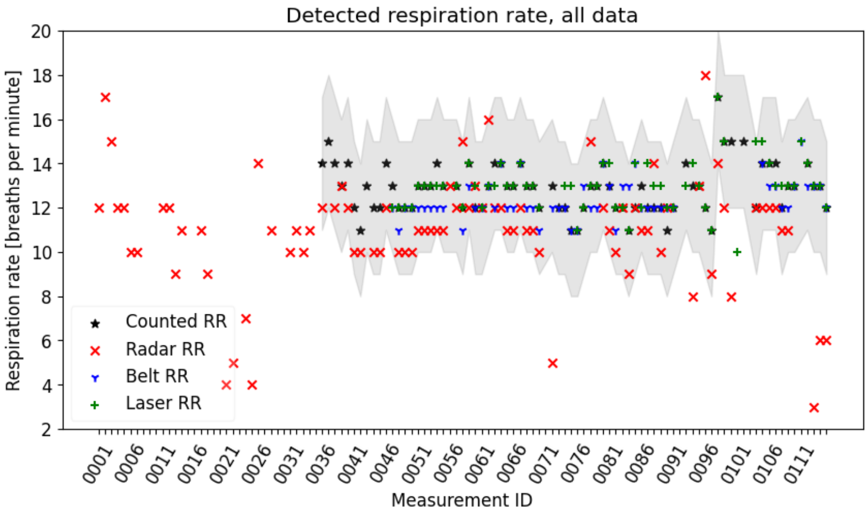


Figure 4.1: Baseline comparison of detected RR with all sensors and all measurements. We show the radar RR (red ×) and if available, counted RR (black ∗), chest belt RR (blue $\curlyvee$), distance laser RR (green +). The gray band denotes a band of ±3 respirations per minute around the counted RR data points.

4.2 Initial get-to-know measurements

Initial set of measurements served to understand what we have at hands, in terms of data acquisition from radar, and the radar signal processing. These measurements are listed in Table 4.1, the *Measurement IDs* from *0001* to *0035*. We performed these measurements only with the radar, we didn't use any other calibration measurement: we didn't use any other sensing method, and we didn't count respirations during the measurement periods.

4.2.1 Experimental setup and environment effects

We performed a series of measurement, during the initial get-to-know phase only with the Fraunhofer FMCW radar. The radar was situated on the ground (together with & in the vicinity of the read out laptop device) whenever the person was laying on the ground e.g. as shown in Fig. 4.2, several meters away from the person.

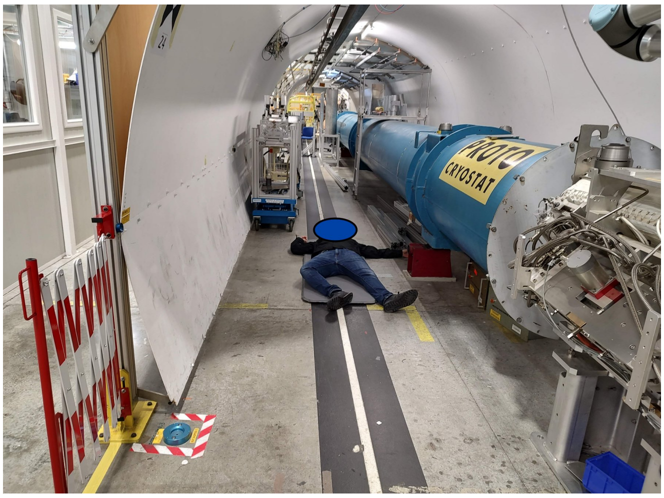


Figure 4.2: A person laying down on their back, with their legs directed towards the observer. The observer may be e.g. a mobile robotic system equipped with a FMCW radar sensor.

Effect of the body position. The person was situated in different positions: sitting down, laying down on side or on back. The person was clothed in a T-shirt or a working jacket. We also attempted to measure angular dependence of RR detection. We tried to understand which are the locations on the person's body towards which we can point the radar and be able to detect RR with it. The most suitable area is the surroundings of the diaphragm, from the front, back and side. We tried also radial direction along legs, towards head/shoulders and confirmed that those directions are not suitable for RR detection, as the diaphragm activity is not detectable there.

Effect of the breathing speed & depth. We tried to understand what breathing patterns we are able to detect, focusing on normal breathing with RR 10-15 breaths per minute, and shalow quick breathing (e.g. while a person

Initial get-to-know measurements

Measurement ID	Comment	
0001	Sitting	
0002		
0003	Sitting, in jacket	
0004		
0005	Sitting, in jacket, from side	
0006	Laying down, on side, facing towards the radar	
0007		
0008	Laying down, on the back	Issue detecting RR (variance)
0009		
0010		
0011	Laying down, on side, back towards the radar	
0012		
0013	Laying down, on side, head radially towards the radar	
0014		
0015	Laying down, on back, shoulder radially towards the radar	Issue detecting RR (variance)
0016		
0017		
0018		
0019	Laying down, on back, legs radially towards the radar	
0020		Radar 20 cm above ground, issue detecting RR (variance)
0021		Radar 20 cm above ground
0022		Radar 20 cm above ground
0023		Radar 40 cm above ground, issue detecting RR (variance)
0024		Radar 40 cm above ground
0025		Radar 40 cm above ground
0026	Sitting, angular dependence	Direct beam (90 °)
0027		80 °, issue detecting RR (var.)
0028		70 °
0029		60 °, issue detecting RR (var.)
0030	Flat field measurement	Radar pointed to the wall
0031	Sitting	
0032		
0033		
0034		
0035		

Table 4.1: Measurements performed to gain initial understanding how the radar works and what to expect.

is performing a cardio-demanding activity) or simulated coughing/gasping (short shallow breaths, e.g. situation prior to cardiac arrest or during seizure). We confirmed that the RR detection works fairly well during the normal breathing activity, and it is difficult and complex (due to volatile variance in the signal) for the shalow quick breathing and coughing/gasping. Similar impossibility to detect RR was observed when the person was moving (radially towards/away from the radar) during the measurement period.

Effect of the clothing materials. We studied effect of clothing and its layering, ranging from a T-shirt, sweater, working jacket. We confirmed that the closer the clothing was to the chest/stomach/diaphragm area, the better chances to detect RR. If we use a clothing layer that does not fit the person – in terms it is much larger and does not touch the diaphragm area – it screens away the respiratory activity and we are not able to detect RR.

Effect of the radar operating temperature. We experimentally found out that for a production use of the radar to detect RR in an industrial settings an external active cooling mechanism would have to be introduced. For the measurement, the radar was connected via a USB port to a read out computer. USB provides both the serial communication and powers the sensor. After approximately 45 minutes of the radar measurement session, the data acquisition ceased to work as the radar overheated (the radar's base was hot, grabbing it by hand was not very comfortable). We had to unplug the radar from the computer for 10-15 minutes and let it cool down, in order to be able to resume measurement session.

4.2.2 Initial understanding summary

During these 35 measurements we gained confidence in the experimental setup, and addressed a variance cut related issue (discussed later in Section 5.3). We then were able to commence measure scenarios outlined in Section 3.4, using multi-sensor measurement, with possibility to calibrate the radar measurements and RR detection with counting, and with sensoric chest wall expansion measurement with a chest belt and remotely with a distance laser.

4.3 Measurements of scenario `M01`

With the `M01` measurement series we attempted to measure 1D spectrum of the radar, to see a possible difference in gain. At first we intended to measure it with a protractor and measure with 10 ° steps in the range of [40 °, 140 °] where 90 ° denotes a direct radial direction towards the subject sternum. With the first 3 measurements ranging from 80 ° to 100 ° we found out that we are not able to realistically distinguish any gain difference with such a small angular step, that our setting is not that precise.

Therefore, with the subject person sitting down (as shown in Fig. 4.3) we embarked on a different approach, where we determined a plane going through person's chest, in the middle between diaphragm plane and shoulder. We noted

Figure 4.3: A person sitting down, facing the observer.

Figure 4.4: Three sensors to detect and calibrate RR: (1) the FMCW Fraunhofer 80 GHz radar, (2) the Bamuer OM70 distance laser, and (3) the PLUX chest belt.

several points in this plane, where zero was on the right side of the person's rib cage, 15 cm on top of the sternum location, and 30cm on the left side of the person's rib cage, moving with 5cm steps.

The first 11 measurements we performed only with the radar, and from the measurement `0047` on we started using also the two calibration measurement sensors, contact-full PLUX chest belt, and contact-less Baumer OM70 distance laser, all three sensors are shown in Fig. 4.4. In addition, the subject person was counting their breaths (in their mind) during the measurement, which provides yet another mean of calibration. In total, we performed 35 measurements, listed in Table 4.2.

`M01` measurements

Measurement ID	Counted RR	Comment
0036	14	90 °, distance ca 1.1 m
0037	15	80 °, distance ca 1.1 m
0038	14	100 °, distance ca 1.1 m
0039	13	0 cm from R side of the person
0040	14	5 cm
0041	12	10 cm
0042	11	15 cm (center of chest)
0043	13	20 cm
0044	12	20 cm again
0045	12	25 cm
0046	14	30 cm (L side of the person)
0047	13	15 cm (center)
0048	12	15 cm (center)
0049	12	10 cm
0050	12	5 cm
0051	13	0 cm
0052	13	20 cm
0053	13	25 cm
0054	14	30 cm
0055	13	15 cm
0056	–	10 cm. Didn't count breaths.
0057	13	10 cm
0058	12	5 cm
0059	14	0 cm
0060	12	15 cm
0061	12	20 cm
0062	13	25 cm
0063	14	30 cm
0064	14	15 cm
0065	13	10 cm
0066	13	5 cm
0067	14	0 cm
0068	13	20 cm
0069	13	25 cm
0070	12	30 cm

Table 4.2: Measurements of scenario `M01`: the person is sitting down, we attempt to measure the angular dependence.

4.3.1 `M01` understanding summary

During these 35 measurements we understood that small-step angular spectrum measurement is not possible due to insufficient precision of the setup. We also performed a series of measurement where we confirmed that pointing the radar (with the distance laser pointer) to different places on person's chest does not lead to degradation in capabilities to detect RR.

4.4 Measurements of scenario `M02`

In the scenario `M02` the subject person is laying down on their back, with their side directed towards the observer, as shown in Fig. 4.5. We performed 8 measurements (listed in Table 4.3) to attempt RR detection from different locations along the person's silhouette.

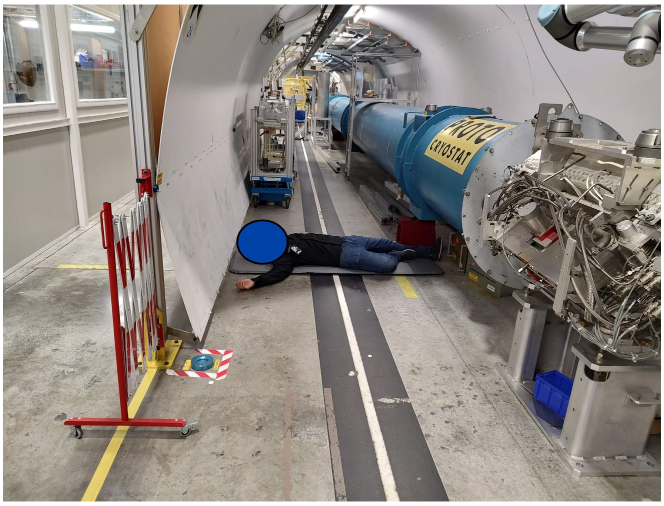


Figure 4.5: A person laying down on their back, with their side directed towards the observer.

`M02` measurements

Measurement ID	Counted RR	Comment
0071		Flat field measurement
0072	13	Pointing to R ear
0073	12	Pointing to R shoulder
0074	12	R shoulder again
0075	11	R chest, bottom of the rib cage
0076	11	R chest (same position)
0077	12	R chest + 5cm up closer to the front chest
0078	13	R waste line
0079	13	R thigh

Table 4.3: Measurements of scenario `M02`: the person is laying on their back.

4.4.1 `M02` understanding summary

During these 8 measurements we confirmed that the most suitable general area from which we are capable detect RR with the baseline model parameters is the diaphragm area and its surrounding (chest, stomach).

4.5 Measurements of scenario `M03`

In the scenario `M03` the subject person is laying down on their side, with their stomach or back directed towards the observer, as shown in Fig. 4.6. We performed 12 measurements (listed in Table 4.4) to attempt RR detection from different locations along the person's rib cage.

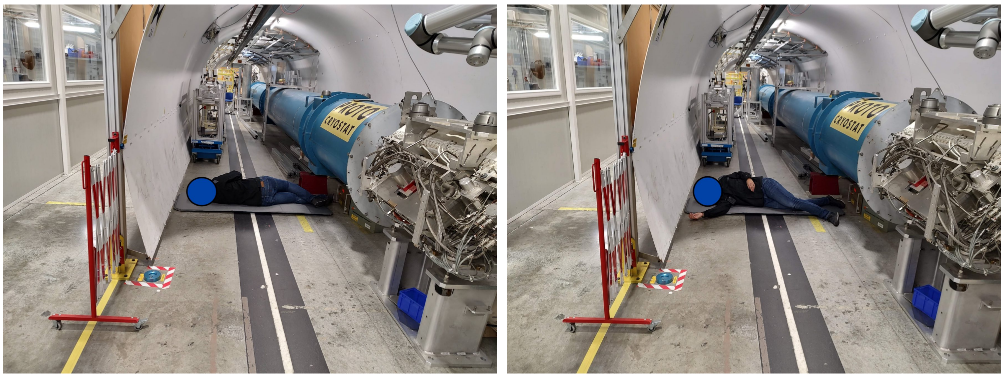


Figure 4.6: A person laying down on their side, facing/rearing the observer.

`M03` measurements

Facing the radar

Measurement ID	Counted RR	Comment
0080	14	R collar bone
0081	13	R chest
0082	12	R rib cage line
0083	12	R waist line
0084	11	R bottom stomach/hips

Back towards the radar

Measurement ID	Counted RR	Comment
0085	12	R back, stomach/hips
0086	13	R back, waist line
0087	12	R back, waist line to rib cage line
0088	12	R back, rib cage line
0089	12	R back, chest
0090	11	R back, scapulas
0091	12	R back, upper shoulders

Table 4.4: Measurements of scenario `M03`: the person is laying down on their side, with their front or back towards the radar.

4.5.1 `M03` understanding summary

During these 12 measurements we confirmed that the diaphragm area and its surrounding (chest, stomach) are great areas to detect RR, and didn't observe any issues with the RR detection.

4.6 Measurements of scenario `M04`

In the scenario `M04` the subject person is laying down on their stomach, with their side directed towards the observer, as shown in Fig. 4.7. We performed 5 measurements (listed in Table 4.5) to attempt RR detection from different locations along the person's rib cage.

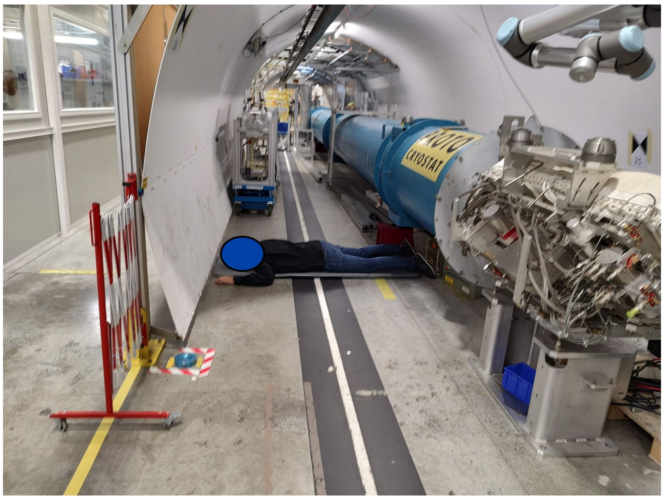


Figure 4.7: A person laying down on their stomach, with their side directed towards the observer.

`M04` measurements

Measurement ID	Counted RR	Comment
0092	–	
0093	14	L shoulders
0094	13	L scapulas
0095	13	L rib cage line
0096	12	L waist line
0097	11	L hips

Table 4.5: Measurements of scenario `M04`: the person is laying on their stomach.

4.6.1 `M04` understanding summary

During these 5 measurements we measured respiratory rate of a person laying down on their stomach. These measurements confirmed that the most reliable detection of RR comes from the diaphragm area and its surrounding (chest, stomach).

4.7 Measurements of scenario `M05`

In the scenario `M05` the subject person is sitting down, wearing a work jacket. The measurements performed are listed in Table 4.6.

`M05` measurements

Measurement ID	Counted RR	Comment
0098	17	center, diaphragm area
0099	15	center, diaphragm area again
0100	15	center, diaphragm, with plexi-glass
0101	–	center, diaphragm, with plexi-glass
0102	15	center, diaphragm, with plexi-glass

Table 4.6: Measurements of scenario `M05`: the person is sitting down, dressed in a work jacket.

4.7.1 `M05` understanding summary

During these 5 measurements we studied how RR is detected when a person wears a work jacket. We concluded that success of RR detection in `M05` measurements depends on how well the jacket fits the person's diaphragm area.

4.8 Measurements of scenario `M06`

In the scenario `M06` the subject person is sitting down and holding a plexi-glass in front in the line of beam, as shown in Fig. 4.8, to study an effect of a possibly transparent barrier. The measurements performed are listed in Table 4.7.

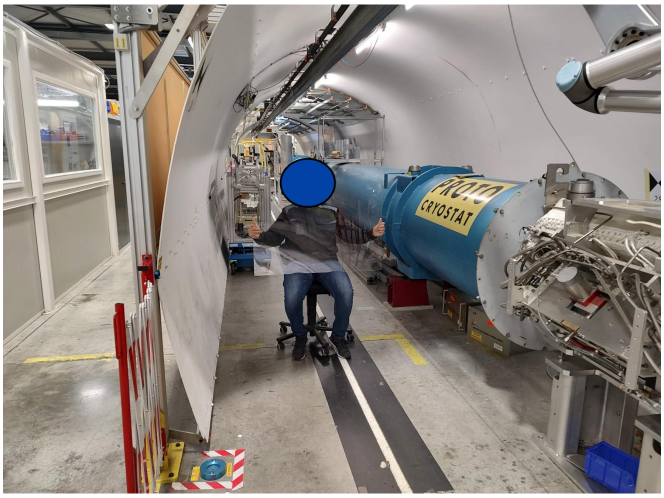


Figure 4.8: A person sitting down, facing the observer, holding a plexi-glass.

4.8.1 `M06` understanding summary

From the 12 `M06` measurements we conclude that the RR detection here is difficult, due to refraction of the plexi-glass, which introduces highly dynamic signal.

`M06` measurements

Measurement ID	Counted RR	Comment
0104	12	
0105	14	
0106	14	
0107	14	
0108	12	
0109	13	
0110	13	Issue detecting RR (variance)
0111	–	Issue detecting RR (variance)
0112	14	Issue detecting RR (variance)
0113	13	
0114	13	
0115	12	

Table 4.7: Measurements of scenario `M06`: the person is sitting down, dressed in a work jacket, with a plexi-glass in front in the line of beam.

5. Radar signal processing

5.1 `HealthDetectionSample` application

CERN Robotic Framework. The CERN Robotics Framework [22] is a modular framework written in C++ for any robotic activity at CERN. One of its modules is the `HealthDetectionSample` application, which encapsulates and connects several other modules for data acquisition (serial communication), and re-usable modules with algorithms, data structures and interfaces for signal processing: e.g. module `Radars` with interfaces for communication with a radar device, `EventLogger` for a generic application-wide logging functionality, `PeakDetection`, `StateEstimator` with Kalman filter, FFT, and the Constant False Alarm Rate (CFAR) [26] algorithm. Its compilation configuration is done with CMake [27], and compilation with `make` [28] with the `gcc` [29] compiler.

CERN Robotic Framework environment. The CERN Robotic Framework applications run on boxes with Ubuntu Linux OS. To follow the computer security guidelines, the OS has to be kept up-to-date with security patches, and from time to time a new long-term support (LTS) version is released, and the CERN Robotic Framework environment has to be brought up to the new standards, making sure all the dependencies suit the dependency versions available in the newer OS. For development and code commissioning purposes it is necessary to have an environment, that 1) is well documented to reproduce, 2) can be accidentally or on purpose trashed, 3) allows for commissioning of the code and the environment (and if needed, versioning of devices tested). In order to benefit from the CERN Robotic Framework package for data acquisition from a FMCW radar and subsequent initial signal processing, we modified a relevant set of modules as follows:

Standalone `HealthDetectionSample` application. In order to document the dependencies, we dissected the `HealthDetectionSample` application from the CERN Robotic Framework codebase. Due to several LTS and minor version OS upgrades, this activity turned out very valuable, by shortening time needed each time to bring the `HealthDetectionSample` environment up to the code. Lastly, we containerized the development & commissioning activity environment, configuring a Docker [30] container with a `Dockerfile` (listed in Section A.3), and running it with Podman [31].

We modified the `HealthDetectionSample` application to allow both the real time data acquisition from the radar, or the ”offline” rerun of the signal processing from the radar raw data files, just with a change of command line input parameters passed to the `HealthDetectionSample` binary.

In order to rapidly prototype possible variations in the parametric study of signal processing parametric model, we implemented `health_detection` application in Python 3, to run the ”offline” signal processing of the radar data. We describe the `health_detection` application in Section 5.4.

5.2 Troubleshooting related to the radar

In this Section we list several aspects of the environment and device setup that we had to take into consideration in the `HealthDetectionSample` application development, to address challenges encountered along the way.

5.2.1 Troubleshooting data acquisition

In Section 4.2.1 we described the effect of the radar operating temperature, where we had to interrupt the measurement session after 45 minutes due to radar overheating.

In addition, initially the `HealthDetectionSample` application expected a manual entry of `SIGTSTP` signal to even start the data acquisition. We have eliminated that need, so that the measurement does not depend on yet another person's reaction time.

5.2.2 Troubleshooting signal processing

To allow evolution of signal processing variations, we introduced an option to choose between data sources for signal processing: either read out real-time data from the radar (and save those in a data file), or read already recorded radar data from a file.

When developing parametric model for RR detection, we encountered several caveats with the signal processing procedure, all related to peak detection approaches. One caveat is related to the variance cut in outliers detection and cleanup: the previously chosen threshold didn't well describe data, therefore we studied variance as one of the important parameters of our RR model.

We encountered a similar issue with peak detection with CFAR algorithm, and besides studying effect of variations of CFAR algorithm parameters on RR detection, we tried also an alternative with Continuous Wavelet Transform (CWT) assisted peak detection. These changes are described in the following Sections.

5.3 Signal processing – cut flow

In this section we describe the radar signal processing cut flow, from the raw data towards detected RR, on example of measurement `0047`. The raw radar data of measurement `0047` is shown in Fig. 5.1. Each measurement, a data frame, contains data lines for `Ramp_Count = 1200` chirps, data line consists of 4000 integers with values in the range of approximately [-4500, +4500].

As it is challenging to visualize dynamics of numbers in Fig. 5.1, we plot also a line-by-line difference of numbers in Fig. 5.2. Let us denote $a[i,j]$ a cell at the i-th row and the j-th column of data matrix of Fig. 5.1, and $b[i,j]$ a cell at the i-th row and the j-th column of data matrix of Fig. 5.2. They are connected as follows:

$$b[i,j] = a[i+1,j] - a[i,j] \tag{5.1}$$

In Fig. 5.3 we plot occurrence distribution histograms of $b[i,j]$ for each data line. There we show also a distribution of max values per data line (black line).

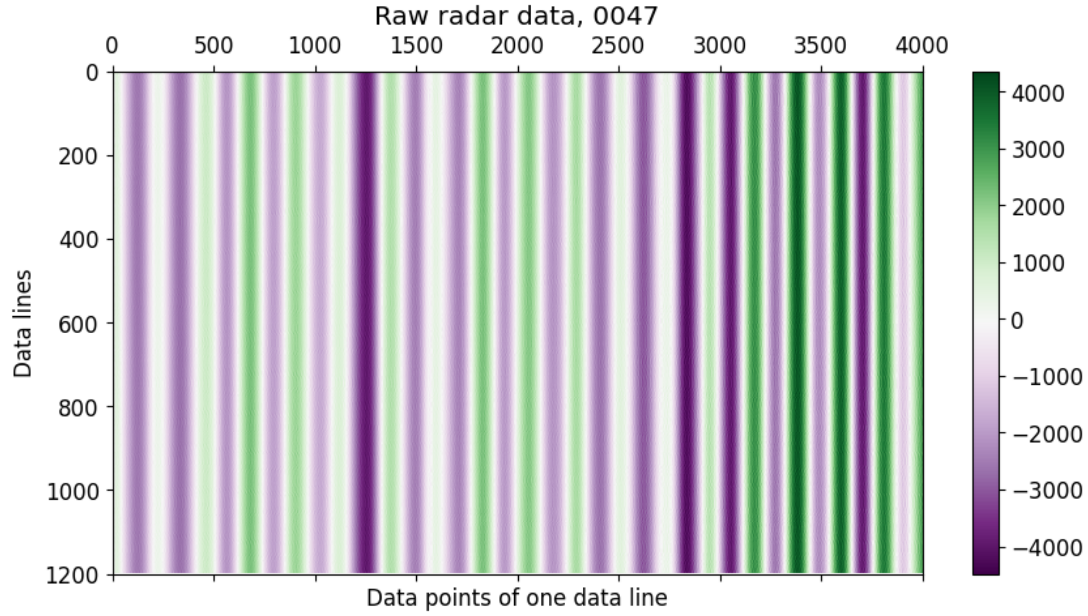


Figure 5.1: Signal processing flow: raw radar data. Each measurement consists of 1200 data lines (`Ramp_Count` radar configuration option), each data line contains 4000 integers. The figure shows that data is consistent among data lines (vertical axis), however there are numerical differences that the visualisation does not show. From these numerical differences the respiratory rate is determined, but they stay hidden to a naked eye.

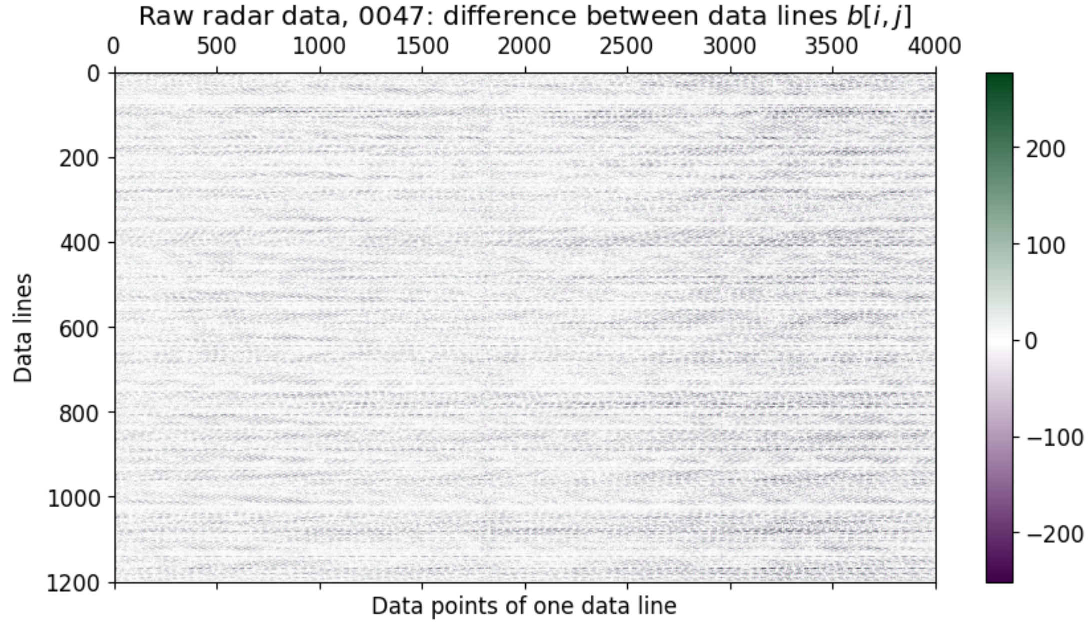


Figure 5.2: Showing difference among radar data lines $b[i,j]$ of Fig. 5.1.

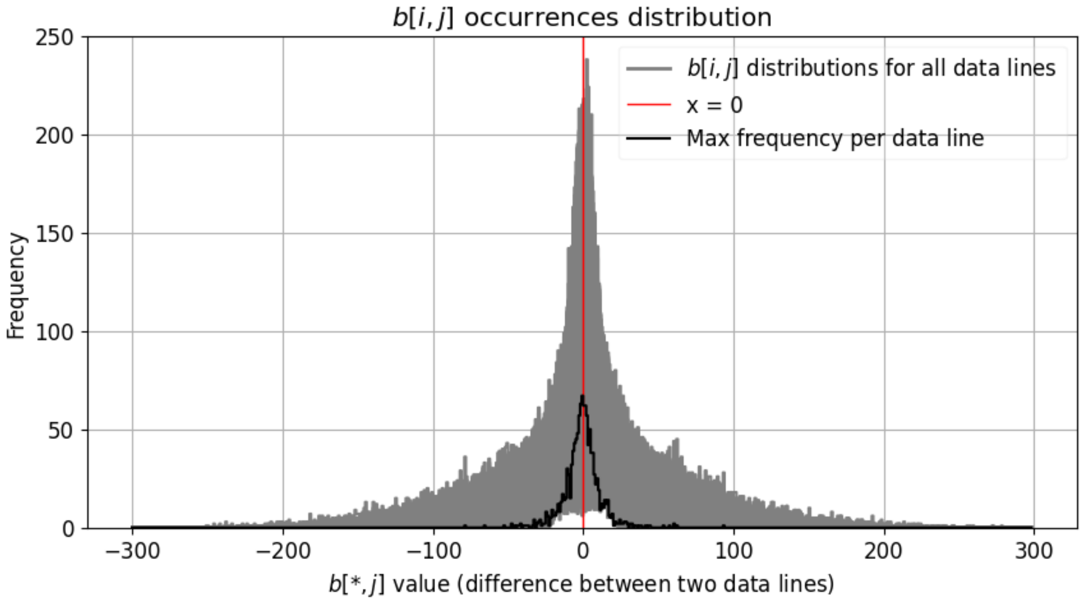


Figure 5.3: Occurrence distribution of $b[i, j]$.

On our way to detect peaks, we apply FFT on the raw data and obtain a range FFT matrix, with the same dimensions as the raw data frame. An example of a data line is shown in Fig. 5.4.

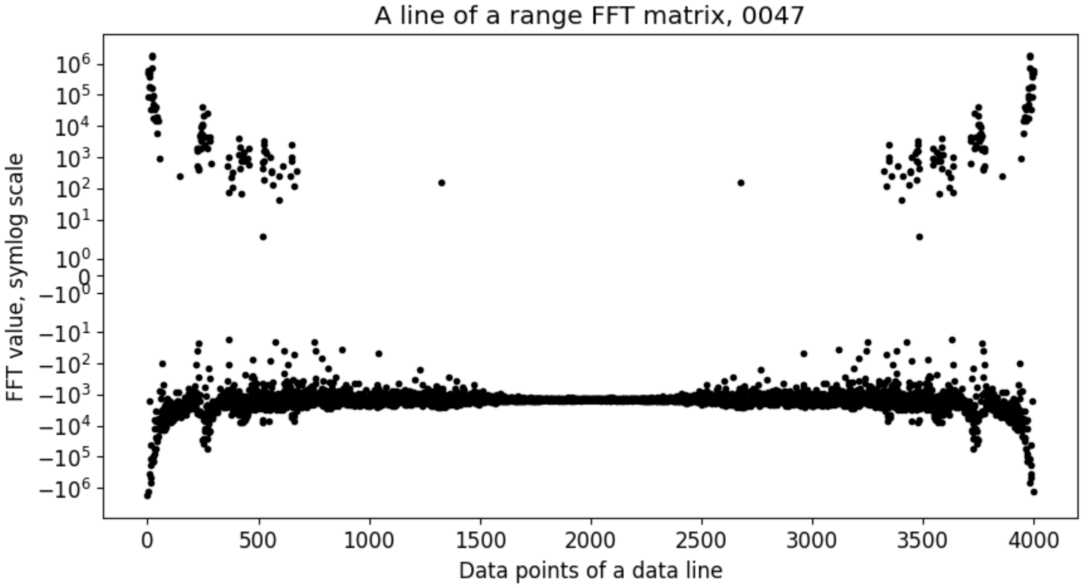


Figure 5.4: Signal processing flow: FFT of raw radar data, a line of a range FFT matrix.

Next we embark onto finding peaks in rows of the range FFT matrix. For that we can use either a CFAR [26] algorithm or the CWT [32] approach. In such a way, for each data line we get indexes of bins of the data line, where peaks are identified, resulting in a peak matrix. Generally, we use only the first element of the peak matrix, reducing it effectively to a peak vector. Several iterations of the peak matrix are shown in Fig. 5.5: We show how the peak matrix looks like after identifying the peaks, then after applying Kalman filter, and then after removing outliers.

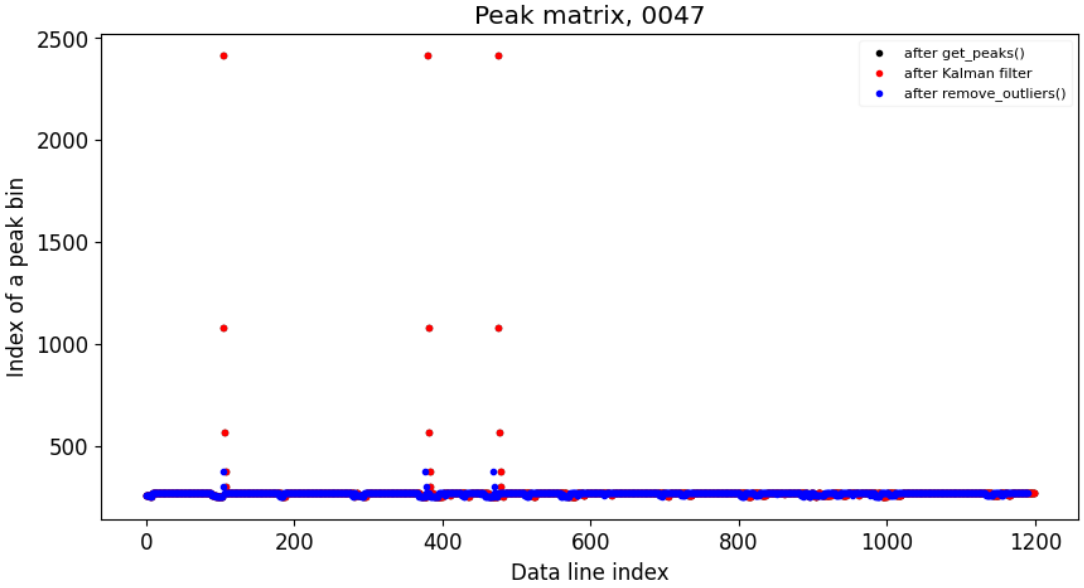


Figure 5.5: Signal processing flow: peak matrix, after `get_peaks()`, after Kalman filter, after `remove_outliers()`.

The CFAR algorithm is used to set a dynamic threshold level to distinguish between signal peaks and noise, based on the statistics of surrounding cells, ensuring a constant false alarm rate. The three parameters we use to configure its output are

- `num_train`: number of training cells used to estimate the noise level. These cells surround the cell under test but exclude the guard cells.
- `num_guard`: number of guard cells around the cell under test, to avoid including the target signal in the noise estimate.
- `false_alarm_rate`: desired false alarm rate, which determines the threshold level.

For each data line of the range FFT matrix, CFAR `find_peaks` algorithm avoids evaluating a window without `num_side` bins on both sides:

$$\begin{aligned}\texttt{num\_side} &= (\texttt{num\_train}\ \mathrm{DIV}\ 2) + (\texttt{num\_guard}\ \mathrm{DIV}\ 2), && (5.2)\\ \text{window} &= \text{data line}[\texttt{num\_side} : \\ &\qquad : \texttt{length of the data line} - \texttt{num\_side}] && (5.3)\end{aligned}$$

For every element in this window, we identify a bin `max_val_pos`, where there is the highest value in the window. For this bin, we calculate a threshold $\alpha * p_{\text{noise}}$, where

$$\begin{aligned}\text{guard\_window\_min} &= \texttt{max\_val\_pos} - (\texttt{num\_guard}\ \mathrm{DIV}\ 2) && (5.4)\\ \text{guard\_window\_max} &= \texttt{max\_val\_pos} + (\texttt{num\_guard}\ \mathrm{DIV}\ 2) && (5.5)\\ \text{guard window} &= \text{data line}[\text{guard\_window\_min} : \\ &\qquad : \text{guard\_window\_max}] && (5.6)\end{aligned}$$

$$\begin{aligned}
\alpha &= \texttt{num\_train} * \left(\texttt{false\_alarm\_rate}^{-1/\texttt{num\_train}} - 1\right) && (5.7)\\
\texttt{sum\_total} &= \sum \text{window} && (5.8)\\
\texttt{sum\_guard} &= \sum \text{guard window} && (5.9)\\
p_{\text{noise}} &= (\texttt{sum\_total} - \texttt{sum\_guard})/\texttt{num\_train} && (5.10)
\end{aligned}$$

When the value in the `max_val_pos` bin is larger than the minimal threshold $\alpha * p_{\text{noise}}$, it is recorded, constructing a peak matrix for all data lines. As for each data line we record only the first bin index, our peak matrix turns effectively into a peak vector, which is an outcome of method `get_peaks()`.

We implemented another peak detection approach with CWT. We parameterize the model with three parameters:

- `scales`: an array of scales at which to perform the wavelet transform. The scales choice can significantly affect the transform's ability to detect features at different sizes and frequencies.
- `wavelet`: type of wavelet to use for the transform.
- `cwt_guard`: number of data points to exclude from the start and end of the data vector during analysis, similar to a guard band in radar processing, it can help reduce edge effects in the wavelet transform.

In order to calculate the CWT coefficients, we use the `PyWavelets` [33] library. To construct the peak matrix with the CWT approach, we use `NumPy`'s [34] `unravel_index` method to identify peaks in the data.

After identifying peaks in `get_peaks()`, if there are peaks detected, we apply a Kalman filter to the peak matrix. There for each element of the peak matrix, we update the state estimate based on the measurement using the Kalman filter's predict and update steps, and then predict the next system state using the system model, returning the current state estimate.

Follows the `remove_outliers()` method to remove the outlier peaks from the detected peaks in the peak matrix, based on variance and a predefined threshold. Calculation of the peak matrix variance goes as follows:

$$\begin{aligned}
\texttt{SUM} &= \sum \texttt{peak\_matrix} && (5.11)\\
\texttt{mean} &= \texttt{SUM}/\text{len}(\texttt{peak\_matrix}) && (5.12)\\
\texttt{square\_sum} &= \sum_{x \in \texttt{peak\_matrix}} \text{int}(x)^2 && (5.13)\\
\texttt{variance} &= \sqrt{\texttt{square\_sum}/\text{len}(\texttt{peak\_matrix}) - \texttt{mean}^2} && (5.14)
\end{aligned}$$

When the calculated `variance` exceeds a threshold `MAX_VARIANCE`, it is not recorded, and we run removal of variance outliers `remove_variance_outliers()`. There we can choose whether we use either `mean` or `median` as a center identification approach, and proceed to remove peak outliers from the peak matrix as follows:

$$\begin{aligned}
\texttt{peak\_matrix} = {} & [\text{int}(\texttt{peak}) \text{ for } \texttt{peak} \text{ in } \texttt{peak\_matrix}\\
& \text{if } (\text{int}(\texttt{peak}) - \texttt{center})/\texttt{variance} \leq\\
& \leq \texttt{OUTLIER\_VARIANCE\_REMOVE\_THRESHOLD}] && (5.15)
\end{aligned}$$

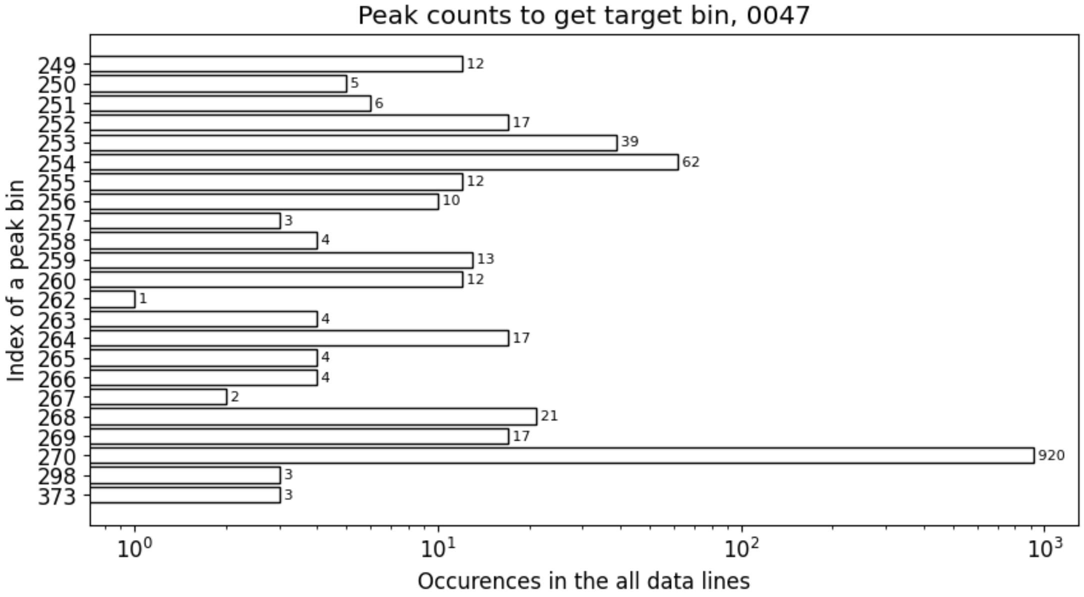


Figure 5.6: Signal processing flow: peak counts to get target bin.

The choice of a `center` method and variance thresholds cuts (`MAX_VARIANCE` and `OUTLIER_VARIANCE_REMOVE_THRESHOLD`), discussed in Section 5.2.2, are paramount in the radar signal processing cut flow, as with too restrictive cutting we were not able to detect any RR in the same data.

We continue to identify the bin (frequency) with the most consistent peak occurrences, which is likely to correspond to the target, in `get_target_bin()`. This procedure is shown in Fig. 5.6.

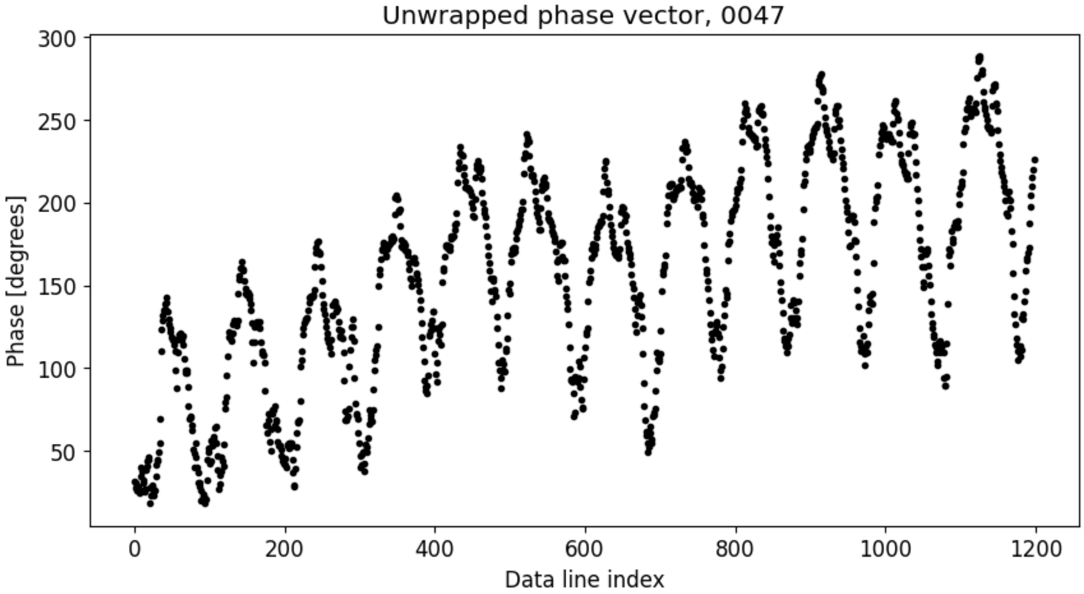


Figure 5.7: Signal processing flow: unwrapped phase vector.

Next, we embark onto unwrapping the phase of the radar signal in `unwrap_phase()` to extract frequency from the data. Example of an unwrapped phase vector is shown in Fig. 5.7.

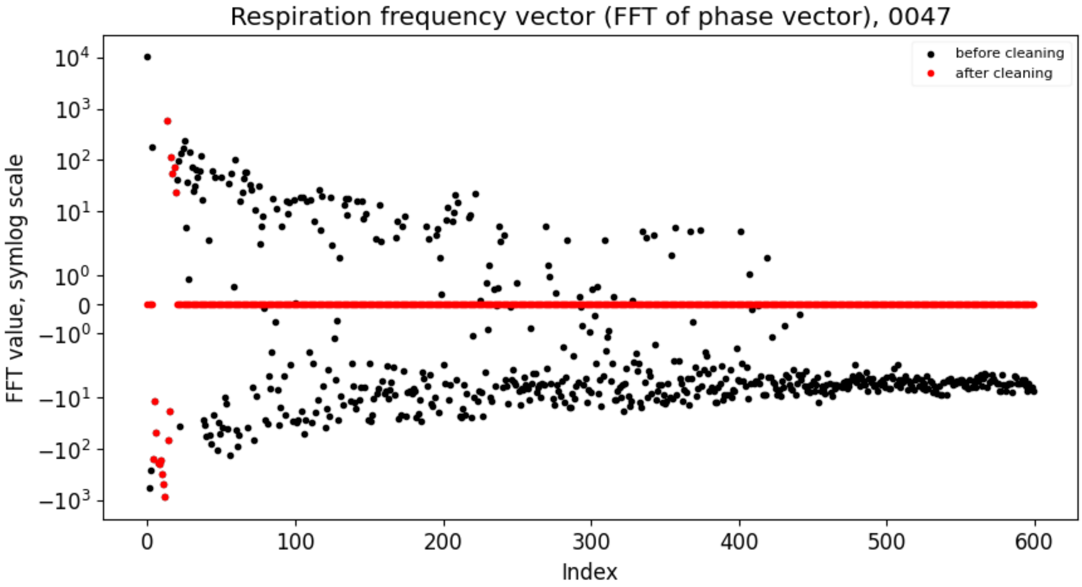


Figure 5.8: Signal processing flow: respiration frequency vector.

Upon applying FFT on the unwrapped phase vector, we obtain a respiration frequency vector, shown in Fig. 5.8, before and after cleanup.

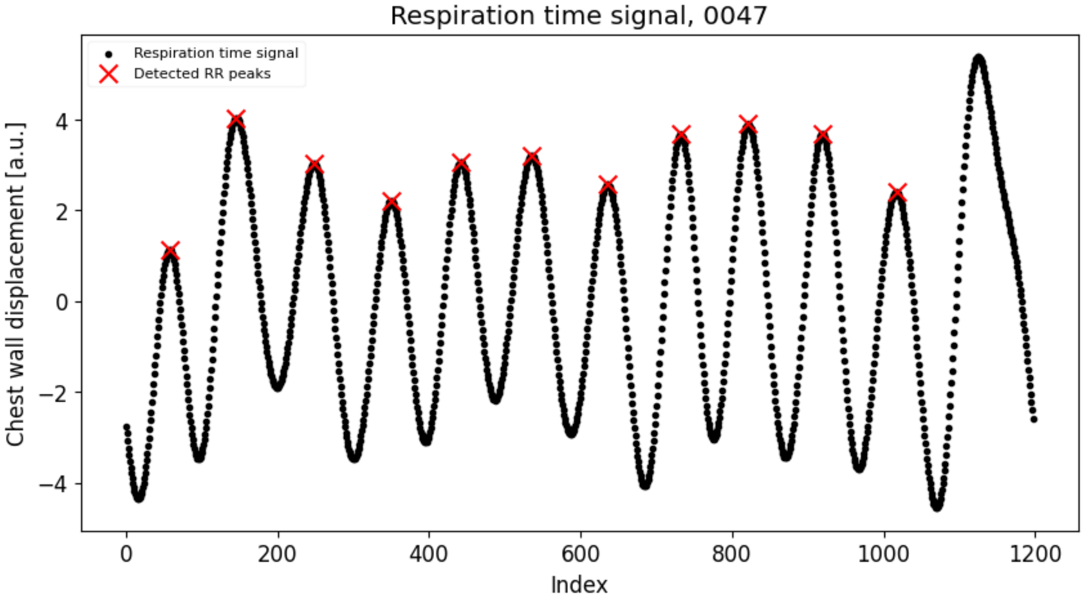


Figure 5.9: Signal processing flow: respiration time signal with detected respiration activity peaks.

Next, we apply inverse FFT on respiration frequency vector, we obtain respiration time signal, as shown in Fig. 5.9. We then use a gradient-based peak detection algorithm to identify respiration peaks in the respiration time signal, the `lung_peaks`: it identifies peaks in a data vector by examining the gradient of the data and selecting points that exceed a specified amplitude threshold as peaks. This method systematically avoids counting in the last peak, as shown in this figure (the red cross is "missing" on top of the last peak), thus systematically, the detected RR may be one breath-per-minute shorter. Using the `lung_peaks` and `unwrapped_phase_vector`, we calculate the detected respiration rate as shown in (5.18).

$$\begin{aligned} L_{\mathrm{lp}} &= \mathrm{len}(\texttt{lung\_peaks}) && (5.16)\\ L_{\mathrm{upv}} &= \mathrm{len}(\texttt{unwrapped\_phase\_vector}) && (5.17)\\ \mathrm{RR} &= \mathrm{int}[L_{\mathrm{lp}}/(0.5 * \texttt{max\_frequency} * L_{\mathrm{upv}}) * 60] && (5.18) \end{aligned}$$

This overview of the cut flow in radar signal processing, from raw radar data frame all the way to a detected respiratory rate, comes handy in the next section, to describe the parametric model parameters variations.

5.4 `health_detection` application for parametric model study

We study a phase space of the parametric model to understand what are the most favorable parameters of the remote detection of human respiratory rate with a Fraunhofer FMCW radar. As shown in this Section 5.3, the RR detection depends on many parameters:

- Peak detection method: either CFAR or CWT.
- CFAR parameters: `num_train`, `num_guard`, `false_alarm_rate`.
- CWT parameters: `num_guard`, max of `scale`, `wavelet`.
- Threshold for gradient detection of RR.
- Threshold for gradient detection of HR, constantly set to 0.1.
- Max variance.
- Thresholds to remove outliers from the peak matrix, and to remove variance outliers; both set to 1.5.
- Threshold to remove variance outliers.
- Size of the Kalman filter initialization sample, constantly set to 10.
- Center methods to remove outliers from the peak matrix, and to remove variance outliers; either `median` or `mean`.

To allow a rapid prototyping of the parameters variations, as well as to remove OS dependencies of the `HealthDetectionSample` application, we implemented `health_detection` application in Python 3. Its code is not listed in this document, however it is a part of the attached archive file, and we describe it in section A.2.4.

We started with machine code translation from C++ to Python 3 with assistance of ChatGPT [35], in order to experiment with and explore the ”new” application of large language models. The machine translation was far from perfect, as we had to solve several issues with code failures, but it was a good start, and we successfully implemented changes to the machine translated code. These changes consisted of e.g. choice of different methods from opensource libraries, different parameters to those methods, or adapting type conversion among various forms of integers and floating point numbers in C++ and their counterparts in Python 3 (which often were not distinguished apart in Python). The final script, the `health_detection` application, does not provide exactly same floating point numbers as the `HealthDetectionSample` application, the difference may be e.g. on 4th or 5th digit after the decimal point, but this precision is perfectly fine for this work, as the final result of the full cut flow provides the same results, within the error of ± 1 breath per minute.

Here we describe elements of `health_detection` application. In Fig. 5.10 we outline the schema of the `health_detection` application. The RR detection flow starts with the `run` function, which serves as a wrapper to initialize instances of all the classes that we need for the radar signal processing:

- `FraunhoferRadar`: managing a radar system configuration;
- `RadarPacket` and `VitalSignalPacket`: data structures to store data about the radar processed signal, and the RR detection results;
- `CFAR` and `CWT`: methods of peak detection;
- `FFT`: a class to calculate FFT of radar data to get a range FFT matrix;
- `DefaultMeasurementModel`: a default measurement model for converting system states into observable measurements;
- `DefaultSystemModel`: a default system model for state estimation purposes;
- `StateEstimatorFilterType`: an enumeration class representing different types of Kalman Filters used in state estimation processes;
- `StateEstimator`: a class that estimates the state of a dynamic system using various types of Kalman Filters. It integrates both system and measurement models to predict and update the state based on incoming measurements;
- `GradientPeakDetection`: implements a gradient-based peak detection algorithm. This class identifies peaks in a data vector by examining the gradient of the data and selecting points that pass a specified amplitude threshold as peaks; and finally,
- `HealthDetection`: a class to process radar signal, detect peak in the radar data and determine the RR.

The `HealthDetection` class provides method `get_vital_signs()`, its schema is shown in Fig. 5.11, which drives the flow of the RR detection:

- First, we read the radar data frame from a file with `load_data()`;

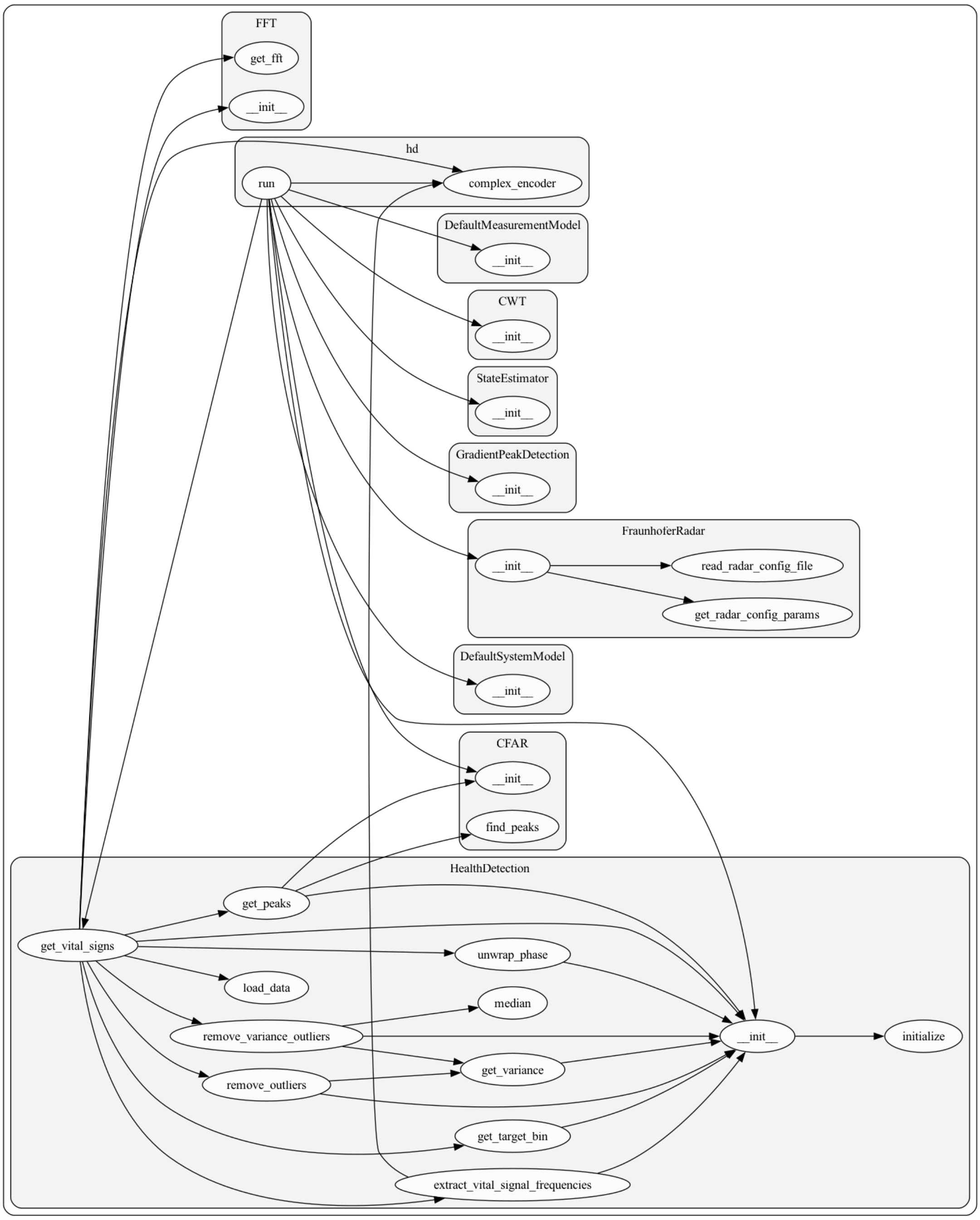


Figure 5.10: Schema of the `health_detection` application.

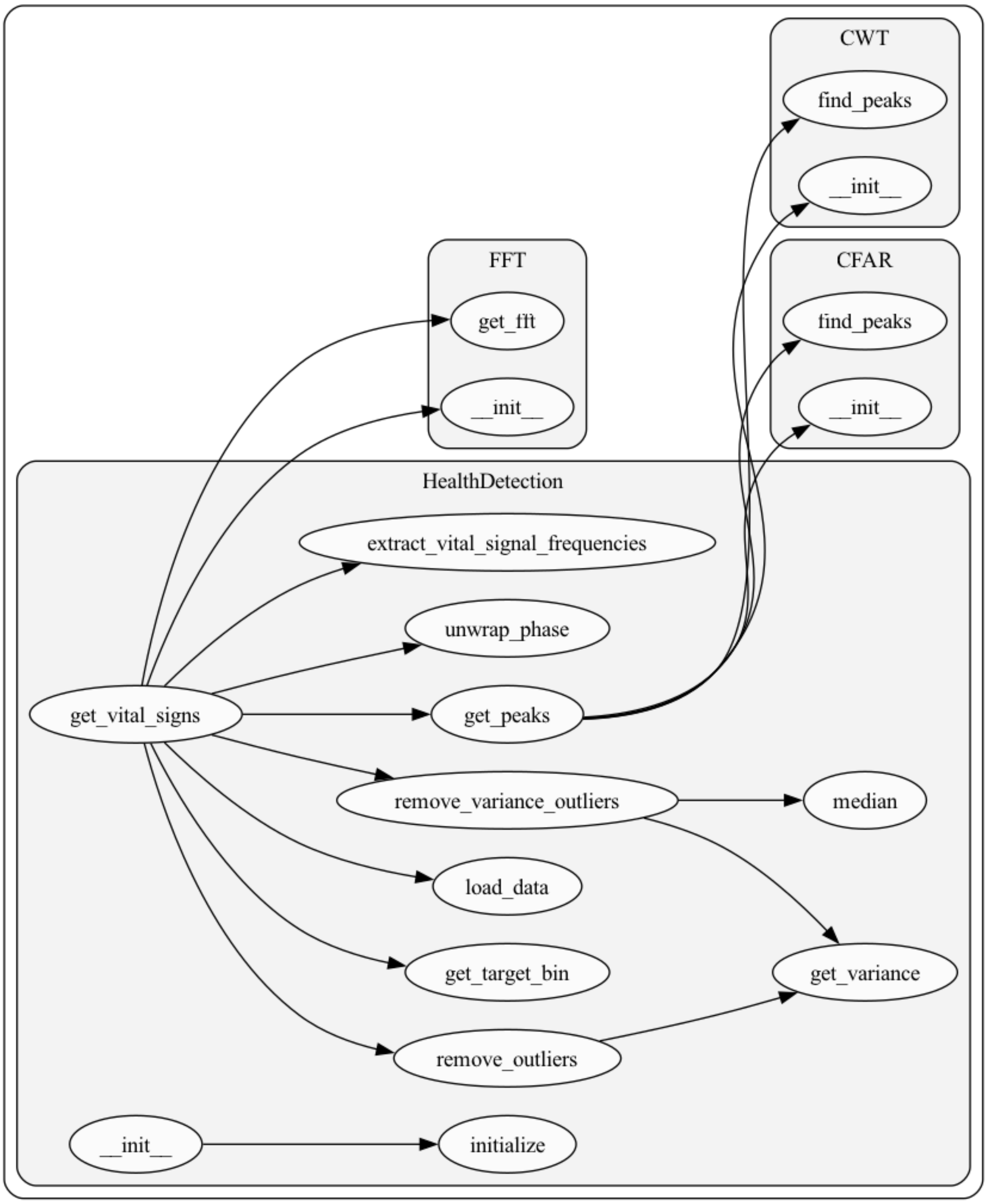


Figure 5.11: Schema of the respiratory rate detection core.

- Then, for every data line, we calculate FFT of the data line with `FFT.get_fft()`;
- We continue with `get_peaks()` to build the peak matrix;
- Next, we apply Kalman filter on the peak matrix;
- and continue with `remove_outliers()` and possibly also `remove_variance_outliers()`, to remove outliers from the peak matrix, or if needed, also remove variance outliers, so that we can continue further with the RR detection;
- We go on to locate bins in the data lines, which correspond to the target location, with `get_target_bin()`, and then unwrap phase with `unwrap_phase()` from the data, to continue further with the vital signs extraction with `extract_vital_signal_frequencies()`;
- There we get respiration frequency vector from unwrapped phase vector through FFT, and gain respiration time signal from it through inverse FFT;
- We then find peaks in the respiration time signal with the `GradientPeakDetection.find_peaks()`;
- And, finally, we calculate RR with (5.18).

We created the code dependencies visualizations from Figs. and with the open source libraries: the offline call graph generator for Python 3 `pyan3` [36], and `Graphviz` an open source graph drawing tools [37].

5.5 Radar respiratory rate detection results

We used the `health_detection` application to scan the parameter space with 51 different parametrizations of the RR detection parametric model. Overview of the parametrizations is listed in Tables 5.1 and 5.2. We started with the parametrization labeled as `BULK_0000`, which clones parametrization of the original model from the `HealthDetectionSample` application. We then started modifying the model parameters, to start detecting RR, and even improve the RR detection. For each parametrization, we ran the RR detection with the full data sample, Measurement IDs in the range [`0001`; `0115`].

A "*TL;DR*" describing Figures 5.12 through 5.15: the desired "ideal look" is a fully green column. Such a column would tell us that for this particular configuration (represented by the column label), the analysis model parameters best suit for the RR detection across different environments and conditions (represented by different measurements in each row). This is not an exact description, however describes the idea behind these plots.

The RR detection results for the first 33 parametrizations (from Table 5.1) are shown in Fig. 5.12 (Measurement IDs in the range [`0001`; `0060`]) and Fig. 5.13 (Measurement IDs in the range [`0061`; `0115`]). The RR results for the last 18 parametrizations (from Table 5.2) are shown in Fig. 5.14 (Measurement IDs in

Label	Peak det.	CFAR		CWT			Thresh.	Max	Center, rm. outliers,	
	method	`#train`	`#guard`	`#guard`	max `scale`	wavelet	∇ RR	variance	peak matrix	variance
BULK_0000	CFAR	50	200				0.5	700.0	mean	median
BULK_0001	CFAR	50	200				0.5	100.0	mean	median
BULK_0002	CFAR	50	200				0.5	6.0	mean	median
BULK_0003	CFAR	10	40				0.5	700.0	mean	median
BULK_0004	CFAR	10	40				0.5	100.0	mean	median
BULK_0005	CFAR	10	40				0.5	6.0	mean	median
BULK_0006	CFAR	80	320				0.5	700.0	mean	median
BULK_0007	CFAR	80	320				0.5	100.0	mean	median
BULK_0008	CFAR	80	320				0.5	6.0	mean	median
BULK_0009	CWT			150	80	gaus1	0.01	700.0	mean	median
BULK_0010	CWT			150	80	gaus1	0.01	100.0	mean	median
BULK_0011	CWT			150	80	gaus1	0.01	6.0	mean	median
BULK_0012	CWT			150	80	cgau1	0.01	700.0	mean	median
BULK_0013	CWT			150	80	cgau1	0.01	100.0	mean	median
BULK_0014	CWT			150	80	cgau1	0.01	6.0	mean	median
BULK_0015	CFAR	50	200				0.5	700.0	mean	mean
BULK_0016	CFAR	50	200				0.5	100.0	mean	mean
BULK_0017	CFAR	50	200				0.5	6.0	mean	mean
BULK_0018	CFAR	50	200				0.5	700.0	median	mean
BULK_0019	CFAR	50	200				0.5	100.0	median	mean
BULK_0020	CFAR	50	200				0.5	6.0	median	mean
BULK_0021	CFAR	50	200				0.5	700.0	median	median
BULK_0022	CFAR	50	200				0.5	100.0	median	median
BULK_0023	CFAR	50	200				0.5	6.0	median	median
BULK_0024	CWT			150	80	gaus1	0.01	700.0	mean	mean
BULK_0025	CWT			150	80	gaus1	0.01	100.0	mean	mean
BULK_0026	CWT			150	80	gaus1	0.01	6.0	mean	mean
BULK_0027	CWT			150	80	gaus1	0.01	700.0	median	mean
BULK_0028	CWT			150	80	gaus1	0.01	100.0	median	mean
BULK_0029	CWT			150	80	gaus1	0.01	6.0	median	mean
BULK_0030	CWT			150	80	gaus1	0.01	700.0	median	median
BULK_0031	CWT			150	80	gaus1	0.01	100.0	median	median
BULK_0032	CWT			150	80	gaus1	0.01	6.0	median	median

Table 5.1: Parametric model scan parameters overview, part 1.

Label	Peak det.	CFAR		Thresh.	Max	Center, rm. outliers,	
	method	`#train`	`#guard`	∇ RR	variance	peak matrix	variance
BULK_0000_C1	CFAR	50	200	0.01	700.0	mean	median
BULK_0001_C1	CFAR	50	200	0.01	100.0	mean	median
BULK_0002_C1	CFAR	50	200	0.01	6.0	mean	median
BULK_0003_C1	CFAR	10	40	0.01	700.0	mean	median
BULK_0004_C1	CFAR	10	40	0.01	100.0	mean	median
BULK_0005_C1	CFAR	10	40	0.01	6.0	mean	median
BULK_0006_C1	CFAR	80	320	0.01	700.0	mean	median
BULK_0007_C1	CFAR	80	320	0.01	100.0	mean	median
BULK_0008_C1	CFAR	80	320	0.01	6.0	mean	median
BULK_0015_C1	CFAR	50	200	0.01	700.0	mean	mean
BULK_0016_C1	CFAR	50	200	0.01	100.0	mean	mean
BULK_0017_C1	CFAR	50	200	0.01	6.0	mean	mean
BULK_0018_C1	CFAR	50	200	0.01	700.0	median	mean
BULK_0019_C1	CFAR	50	200	0.01	100.0	median	mean
BULK_0020_C1	CFAR	50	200	0.01	6.0	median	mean
BULK_0021_C1	CFAR	50	200	0.01	700.0	median	median
BULK_0022_C1	CFAR	50	200	0.01	100.0	median	median
BULK_0023_C1	CFAR	50	200	0.01	6.0	median	median

Table 5.2: Parametric model scan parameters overview, part 2.

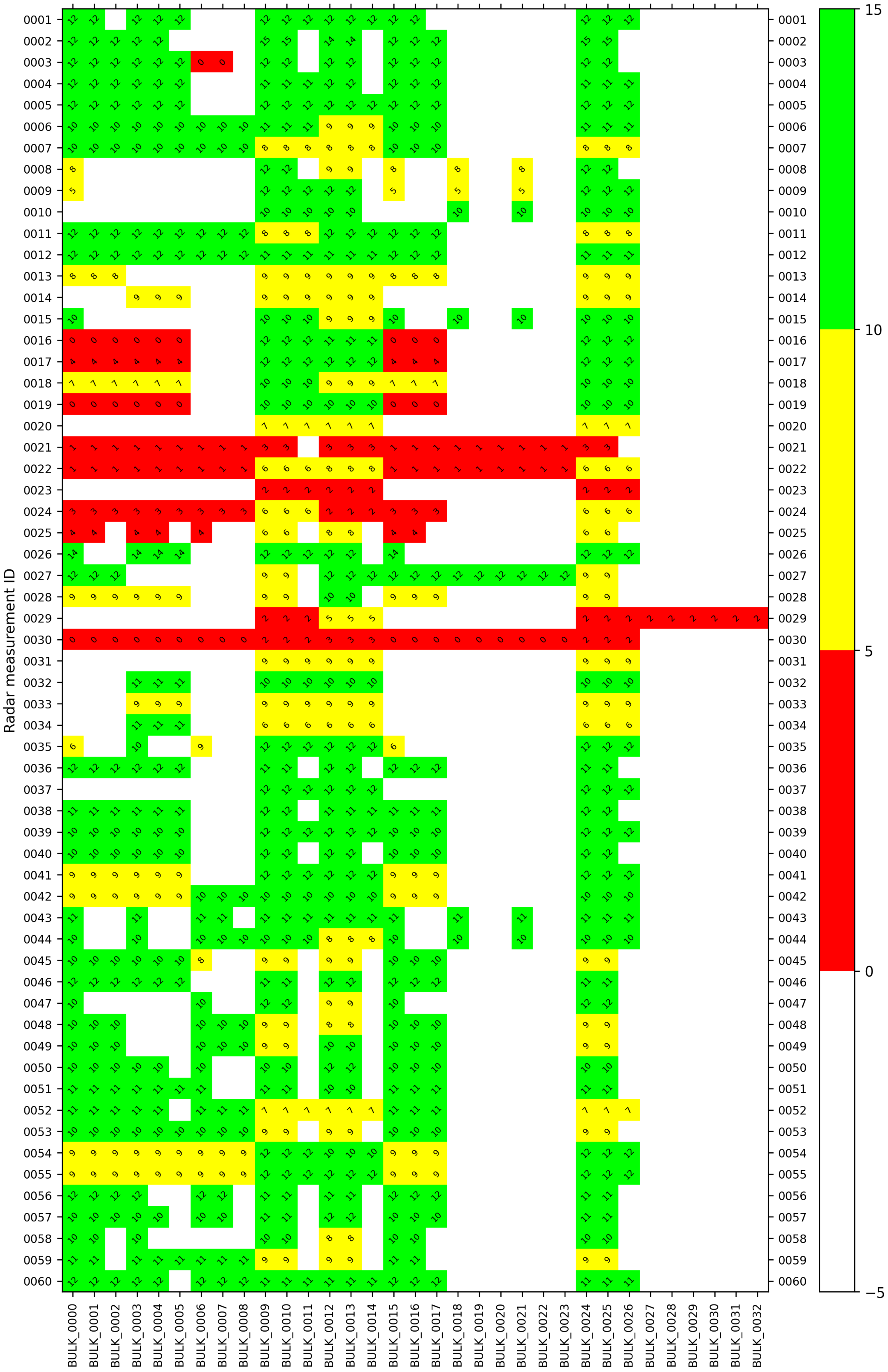


Figure 5.12: Results of respiratory rate detection, part 1.

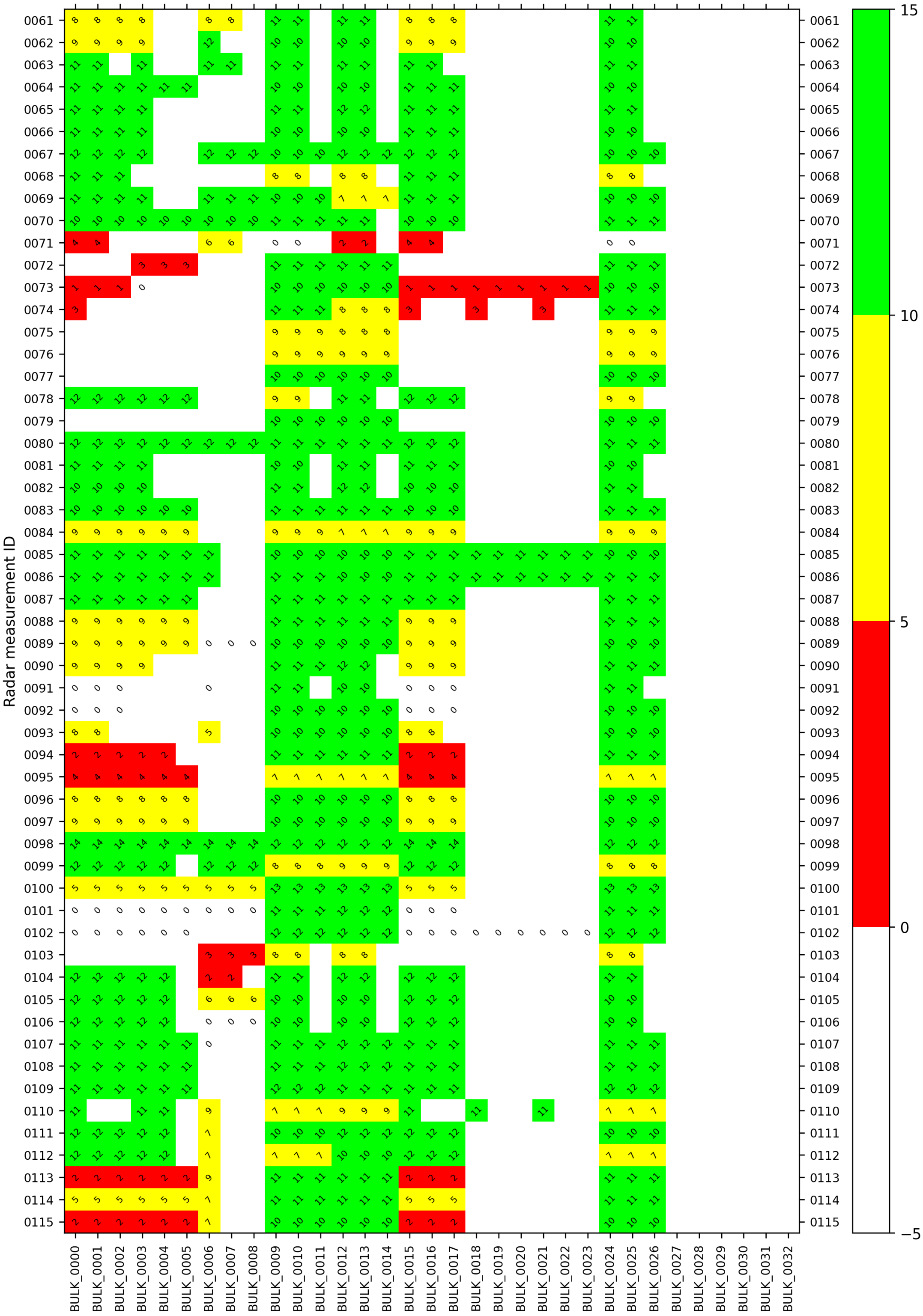
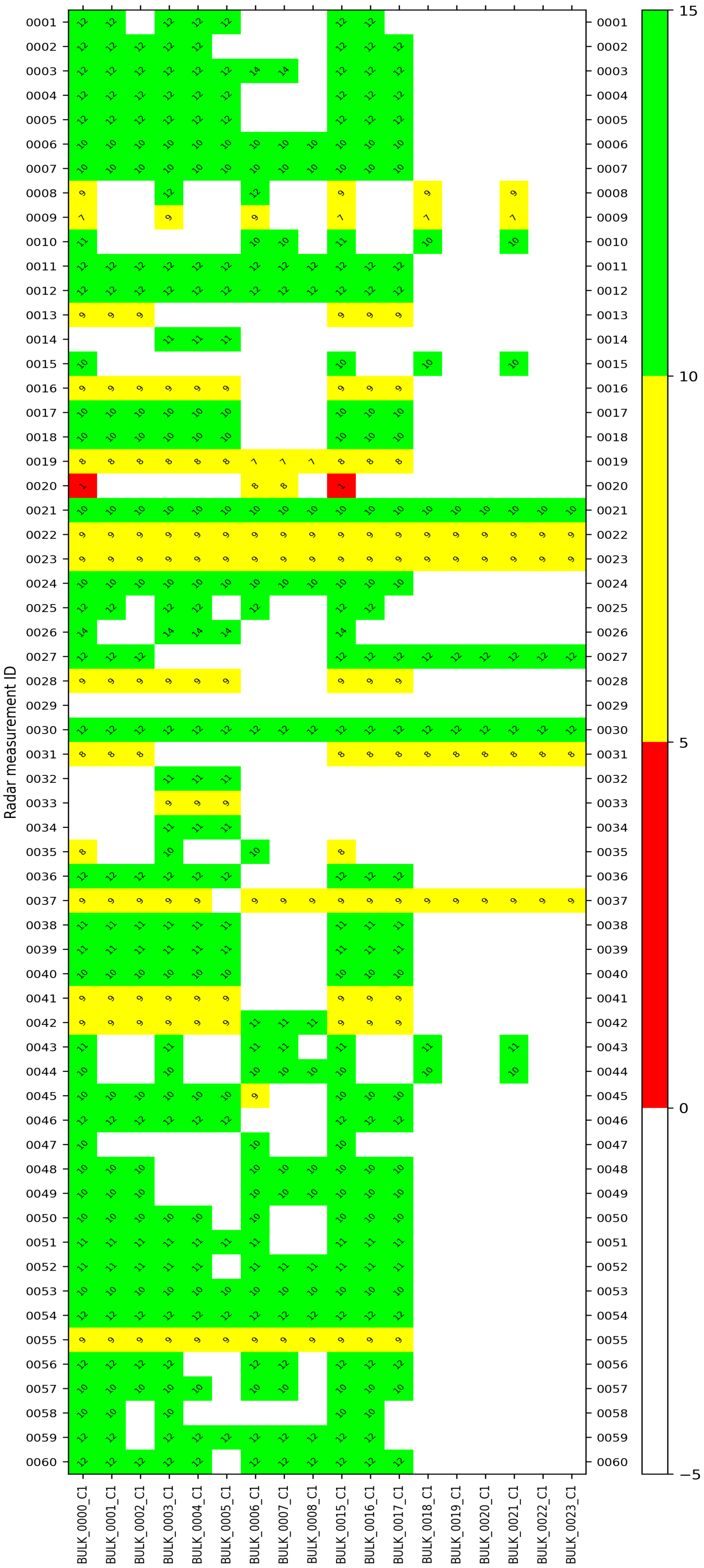


Figure 5.13: Results of respiratory rate detection, part 2.



Radar measurement ID	BULK_0000_C1	BULK_0001_C1	BULK_0002_C1	BULK_0003_C1	BULK_0004_C1	BULK_0005_C1	BULK_0006_C1	BULK_0007_C1	BULK_0008_C1	BULK_0015_C1	BULK_0016_C1	BULK_0017_C1	BULK_0018_C1	BULK_0019_C1	BULK_0020_C1	BULK_0021_C1	BULK_0022_C1	BULK_0023_C1
0001	12	12		12	12	12				12	12							
0002	12	12	12	12	12					12	12	12						
0003	12	12	12	12	12	12	14	14		12	12	12						
0004	12	12	12	12	12	12				12	12	12						
0005	12	12	12	12	12	12				12	12	12						
0006	10	10	10	10	10	10	10	10	10	10	10	10						
0007	10	10	10	10	10	10	10	10	10	10	10	10						
0008	9			12			12			9			9			9		
0009	7			9			9			7			7			7		
0010	11						10	10		11			10			10		
0011	12	12	12	12	12	12	12	12	12	12	12	12						
0012	12	12	12	12	12	12	12	12	12	12	12	12						
0013	9	9	9							9	9	9						
0014				11	11	11												
0015	10									10			10			10		
0016	9	9	9	9	9	9				9	9	9						
0017	10	10	10	10	10	10				10	10	10						
0018	10	10	10	10	10	10				10	10	10						
0019	8	8	8	8	8	8	7	7	7	8	8	8						
0020	1						8	8		1								
0021	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
0022	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
0023	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
0024	10	10	10	10	10	10	10	10	10	10	10	10						
0025	12	12		12	12		12			12	12							
0026	14			14	14	14				14								
0027	12	12	12							12	12	12	12	12	12	12	12	12
0028	9	9	9	9	9	9				9	9	9						
0029																		
0030	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
0031	8	8	8							8	8	8	8	8	8	8	8	8
0032				11	11	11												
0033				9	9	9												
0034				11	11	11												
0035	8			10			10			8								
0036	12	12	12	12	12	12				12	12	12						
0037	9	9	9	9	9		9	9	9	9	9	9	9	9	9	9	9	9
0038	11	11	11	11	11	11				11	11	11						
0039	11	11	11	11	11	11				11	11	11						
0040	10	10	10	10	10	10				10	10	10						
0041	9	9	9	9	9	9				9	9	9						
0042	9	9	9	9	9	9	11	11	11	9	9	9						
0043	11			11			11	11		11			11			11		
0044	10			10			10	10	10	10			10			10		
0045	10	10	10	10	10	10	9			10	10	10						
0046	12	12	12	12	12	12				12	12	12						
0047	10						10			10								
0048	10	10	10				10	10	10	10	10	10						
0049	10	10	10				10	10	10	10	10	10						
0050	10	10	10	10	10		10			10	10	10						
0051	11	11	11	11	11	11	11			11	11	11						
0052	11	11	11	11	11		11	11	11	11	11	11						
0053	10	10	10	10	10	10	10	10	10	10	10	10						
0054	12	12	12	12	12	12	12	12	12	12	12	12						
0055	9	9	9	9	9	9	9	9	9	9	9	9						
0056	12	12	12	12			12	12		12	12	12						
0057	10	10	10	10	10		10	10		10	10	10						
0058	10	10		10						10	10							
0059	12	12		12	12	12	12	12	12	12	12							
0060	12	12	12	12	12		12	12	12	12	12	12						

Figure 5.14: Results of respiratory rate detection, part 3.

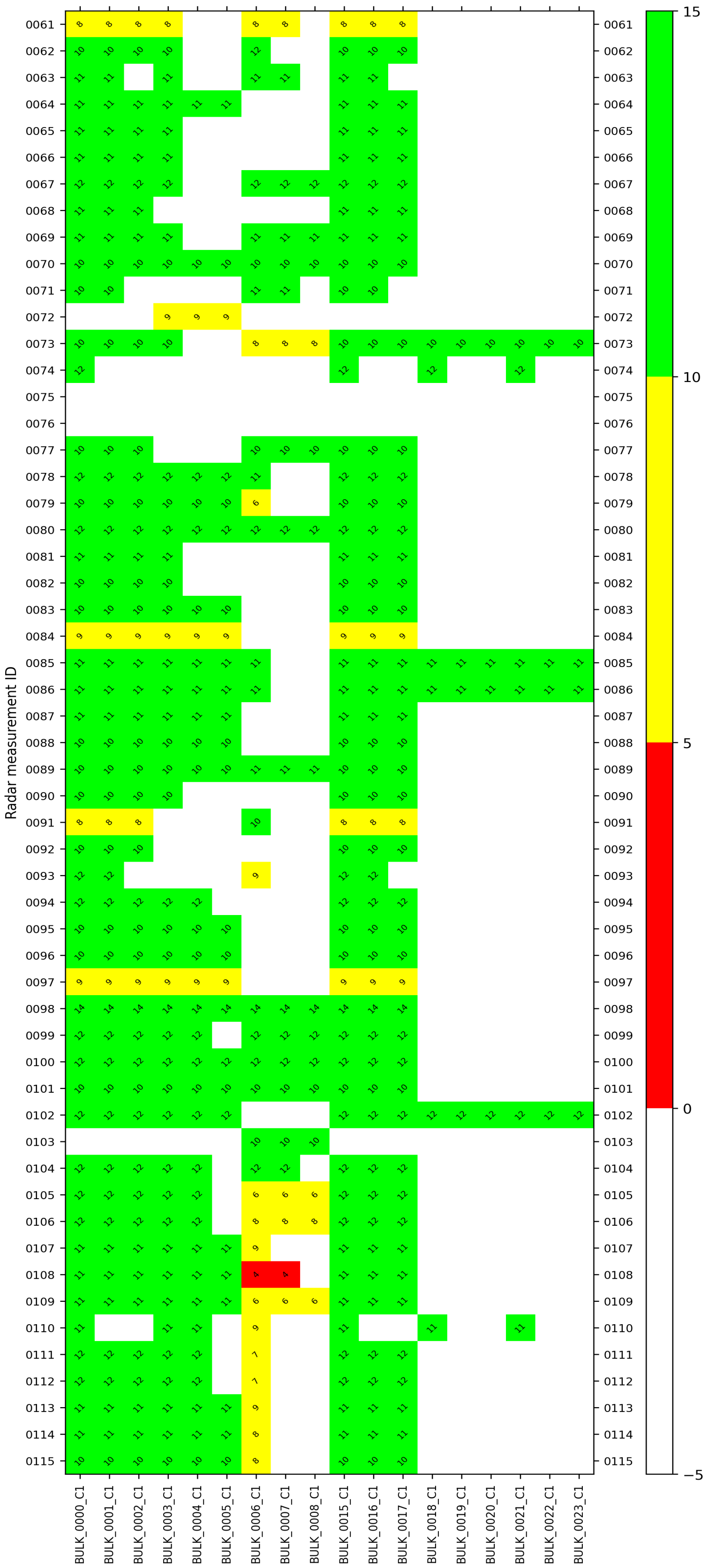


Figure 5.15: Results of respiratory rate detection, part 4.

the range [`0001`; `0060`]) and Fig. 5.15 (Measurement IDs in the range [`0061`; `0115`]).

These four figures, Fig. 5.12 through Fig. 5.15 show a heatmap of detected RR with different parametrization. The heatmap rows corresponds to the same data file, while a column corresponds to a same parametrization. In these figures, the white background corresponds to failure to detect a respiratory rate (or the detected RR was zero breaths per minute). The green background color corresponds to detected RR in the range of [10, 15] breaths per minute, which is an ideal RR range. The yellow background color corresponds to detected RR in the range of (5, 10), while red background color corresponds to the detected RR in the range of (0, 5]; both are suspiciously low. In the cells where RR detection revealed a RR, we print the RR in breaths per minute.

5.5.1 Radar RR detection results with CFAR peak detection model

The baseline parametrization, `BULK_0000`, corresponds to the very original parametrization with CFAR peak detection method. As shown in the first two figures, it does not lead to RR detection for the full sample of 115 measurements, thus we tried different parametrizations to find a more suitable set.

At first, we stick to the CFAR peak detection method, and tried to variate the CFAR model parameters, as well as the max variance, freezing the center method for both outliers removals, in parametrizations `BULK_0001` through `BULK_0008`. The parametrizations `BULK_0006` through `BULK_0008` do not seem to have improved the RR detection over the baseline `BULK_0000`.

The parametrizations `BULK_0015` through `BULK_0023` used CFAR model, and focused on variation of the center method for both outliers removals. The first three parametrizations from this set slightly improved the situation over the parametrizations from the previous paragraph, but the rest, `BULK_0018` through `BULK_0023` made it even worse (large amount of white background cells).

The last set of experiments we performed for parametrizations with CFAR are those listed in Table 5.2. Parametrizations `BULK_0018_C1` through `BULK_0023_C1` do not serve well at all to detect RR. They all have one parameter in common: the center method for peak matrix outliers removal is `median`, similarly for `BULK_0018` through `BULK_0023`. The parametrizations `BULK_0006_C1` through `BULK_0008_C1` are slightly better, but still not resilient. They all have `num_train`=80 and `num_guard`=320, same as `BULK_0006` through `BULK_0008`.

In Table 5.3 we sort parametrizations by the number of green, yellow, red, and white cells. From this table it seems that the parametrizations `BULK_0000_C1` and `BULK_0015_C1` provide the most suitable parametrization for RR detection with

Label	# green	# yellow	# red	# white
BULK_0000_C1	87	18	1	9
BULK_0015_C1	87	18	1	9
BULK_0003_C1	80	15	0	20
BULK_0001_C1	79	15	0	21
BULK_0016_C1	79	15	0	21
BULK_0002_C1	72	15	0	28
BULK_0017_C1	72	15	0	28
BULK_0000	59	21	12	23
BULK_0015	59	21	12	23
BULK_0004_C1	64	13	0	38
BULK_0003	56	18	10	31
BULK_0001	53	18	11	33
BULK_0016	53	18	11	33
BULK_0002	49	17	9	40
BULK_0017	49	17	9	40
BULK_0006_C1	46	22	1	46
BULK_0005_C1	50	12	0	53
BULK_0004	44	15	10	46
BULK_0005	32	15	8	60
BULK_0007_C1	35	11	1	68
BULK_0006	28	15	6	66
BULK_0008_C1	27	9	0	79
BULK_0007	22	6	5	82
BULK_0008	18	4	4	89
BULK_0018_C1	13	6	0	96
BULK_0021_C1	13	6	0	96
BULK_0018	8	2	4	101
BULK_0021	8	2	4	101
BULK_0019_C1	7	4	0	104
BULK_0020_C1	7	4	0	104
BULK_0022_C1	7	4	0	104
BULK_0023_C1	7	4	0	104
BULK_0019	3	0	3	109
BULK_0020	3	0	3	109
BULK_0022	3	0	3	109
BULK_0023	3	0	3	109

Table 5.3: Radar RR detection results, CFAR models.

CFAR peak detection model:

$$\left\{\begin{array}{lcl}
\texttt{num\_train} & = & 50 \\
\texttt{num\_guard} & = & 200 \\
\texttt{false\_alarm\_rate} & = & 0.001 \\
\text{max variance} & = & 700 \\
\text{Thresh. } \nabla \text{ RR} & = & 0.01 \\
\text{Thresh. } \nabla \text{ HR} & = & 0.1 \\
\text{Thresh. to remove peak matrix outliers} & = & 1.5 \\
\text{Thresh. to remove variance outliers} & = & 1.5 \\
\text{Kalman filter initialization sample size} & = & 10 \\
\text{Center for peak matrix outlier removal} & = & \texttt{mean}
\end{array}\right. \tag{5.19}$$

With the `center` for variance outlier removal method being

- `median` for `BULK_0000_C1`,
- `mean` for `BULK_0015_C1`.

5.5.2 Radar RR detection results with CWT peak detection model

In case of parametrizations with CWT peak detection, we chose nicely the CWT model parameters, thus we embarked on scanning a phase space with variating the max variance and the center methods for both outlier removals. We ran the scan with 15 different parametrizations with CWT peak detection (shown in Table 5.1).

Using the same comparison method for parametrizations with CWT peak detection model, the CWT parametrization results are shown as a part of Fig. 5.12 and 5.13.

The most suitable parametrizations for RR detection with CWT peak detection model, as shown in Table 5.4, are `BULK_0009`, `BULK_0010`, `BULK_0024`, and `BULK_0025`, with parametrization

$$\left\{\begin{array}{lcl}
\texttt{cwt\_guard} & = & 150 \\
\texttt{scales} & = & 80 \\
\texttt{wavelet} & = & \texttt{gaus1} \\
\text{Thresh. } \nabla \text{ RR} & = & 0.01 \\
\text{Thresh. } \nabla \text{ HR} & = & 0.1 \\
\text{Thresh. to remove peak matrix outliers} & = & 1.5 \\
\text{Thresh. to remove variance outliers} & = & 1.5 \\
\text{Kalman filter initialization sample size} & = & 10 \\
\text{Center for peak matrix outlier removal} & = & \texttt{mean}
\end{array}\right. \tag{5.20}$$

With the max variance being

- 700 for `BULK_0009` and `BULK_0024`,
- 100 for `BULK_0010` and `BULK_0025`,

and the `center` for variance outlier removal method being

- `median` for `BULK_0009` and `BULK_0010`,
- `mean` for `BULK_0009` and `BULK_0010`.

Label	# green	# yellow	# red	# white
BULK_0009	81	29	4	1
BULK_0010	81	29	4	1
BULK_0024	81	29	4	1
BULK_0025	81	29	4	1
BULK_0012	78	32	5	0
BULK_0013	78	32	5	0
BULK_0011	56	18	3	38
BULK_0026	56	18	3	38
BULK_0014	48	22	4	41
BULK_0027	0	0	1	114
BULK_0028	0	0	1	114
BULK_0029	0	0	1	114
BULK_0030	0	0	1	114
BULK_0031	0	0	1	114
BULK_0032	0	0	1	114

Table 5.4: Radar RR detection results, CWT models.

5.6 Preparing for calibration data processing

In Section 3.2.1 we discussed how we perform simultaneous measurement and detection of respiratory activity with a loosely coupled system of 3 sensors (radar, chest belt, laser), and how we do not need to apply complex distributed systems algorithms to assure time synchronization in the system, because it is sufficient to identify a 1-minute interval when the measurement is performed.

At first, we identified start and end timestamps for each data taking 1-minute interval: for each measurement we have an application log from the `HealthDetectionSample` application (Fig. 5.16), where we chose to print highest level of logs to have debug information available to understand all aspects of radar data acquisition and processing.

From this application log we detected the first and last timestamps of of the `SerialCommunication` module log with message denoting the full data frame being read out from the radar, the `Number of bytes read:  4095` message.

Then we used these interval end points to map this interval to data series of chest belt and laser, allowing us process the calibration sensors data independently. Code snippets relevant to this process are listed in Section A.2.1.

```
[X3:53.051] [HealthDetection] [info] STARTING
[X3:53.051] [FraunhoferRadar] [debug] initialize
[X3:53.051] [SerialCommunication] [info] initialize
[X3:53.051] [SerialCommunication] [info]
    50 deviceName_ /dev/ttyACM0 fileFlags 2306
[X3:53.051] [FraunhoferRadar] [debug] radarConfiguration
[X3:53.051] [FraunhoferRadar] [debug]
   calculateRadarParameters
...
[X3:53.058] [FraunhoferRadar] [info] Radar initialization
   completed.
[X3:53.058] [FraunhoferRadar] [debug]
   getMaxObservationFrequency
[X4:03.058] [FraunhoferRadar] [debug] getFrame
[X4:03.058] [FraunhoferRadar] [debug] startRamp
[X4:03.059] [SerialCommunication] [debug]
    Number of bytes read: 1
...
[X4:03.165] [SerialCommunication] [debug]
    Number of bytes read: 4095
...
[X5:07.990] [SerialCommunication] [debug]
    Number of bytes read: 4095
[X5:07.990] [SerialCommunication] [debug]
    Number of bytes read: 1320
...
[X5:10.142] [HealthDetection] [debug] respirations: 11,
   heartbeats: 69
[X5:10.142] [Radar Sample] [info] RR: 10 bpm, HR:
[X5:10.142] [HealthDetection] [debug] deinitialize
[X5:10.142] [FraunhoferRadar] [debug] deinitialize
[X5:10.142] [SerialCommunication] [info] deinitialize
```

Figure 5.16: Example of the `HealthDetectionSample` application log (`M0048`). The block highlighted in blue shows the endpoints of the active radar data taking period.

6. Chest belt signal processing

In this chapter we describe signal processing from the PLUX chest belt sensor (Section 3.2). First, we describe data acquisition and preparation of data for processing in Section 6.1, and the respiratory rate detection approach in Section 6.2.

6.1 Data acquisition with PLUX chest belt

We used the PLUX chest belt as an independent contact-full calibration sensor in addition to RR detection by the radar. During the data taking period, the subject person was wearing the chest belt, data was acquired, and read out via a Bluetooth connection, using a Python script that leverages PLUX-provided closed source libraries for data acquisition. The chest belt data acquisition ran through the whole session with multiple radar measurements, therefore we used procedure described in Section 5.6 to identify the periods when respiration rate was detected by radar.

```
Time stamp, global_resp, global_ecg
WeekDay Month Day X4:05 YYYY
    ,-0.3173037308308538,1.4720314033366044,
WeekDay Month Day X4:05 YYYY
    ,-0.3167544060425727,1.4720314033366044,
... ### another 1248 rows with the exact same timestamp
```

Figure 6.1: Example of the PLUX chest belt data series, before the timestamp correction.

Upon identification of the radar measurement periods in chest belt data, we noticed that the chest belt data contains a series of 1250 data points with the exact same timestamp string, spaced 5 s between two clusters. Each data point (example shown in Fig. 6.1) consists of a timestamp, the distance measurement (chest wall displacement) in arbitrary units [a. u.], and a dummy column of a constant value, a placeholder for possible heart rate detection that we didn't pursue as it goes beyond objectives of this work.

```
Time stamp, global_resp, global_ecg
[X4:05.0],-0.3173037308308538,1.4720314033366044,
[X4:05.004],-0.3167544060425727,1.4720314033366044,
... ### another 1248 rows with the exact same timestamp
```

Figure 6.2: Example of the PLUX chest belt data series, after the timestamp correction.

Inspection of the raw data revealed a pattern in the data files: there are consistently 1250 data points with the exact same timestamp string, different one every 5 seconds. Thus we applied a time correction to the data points: For each data point in a cluster of 1250 subsequent data points, we incremented the timestamp by 4 ms with respect to the previous data point timestamp, as shown in Fig. 6.2. When we plotted the original raw data of *[timestamp, distance]*, it looked as a series of time-equi-distant spikes, as shown in Fig. 6.3.

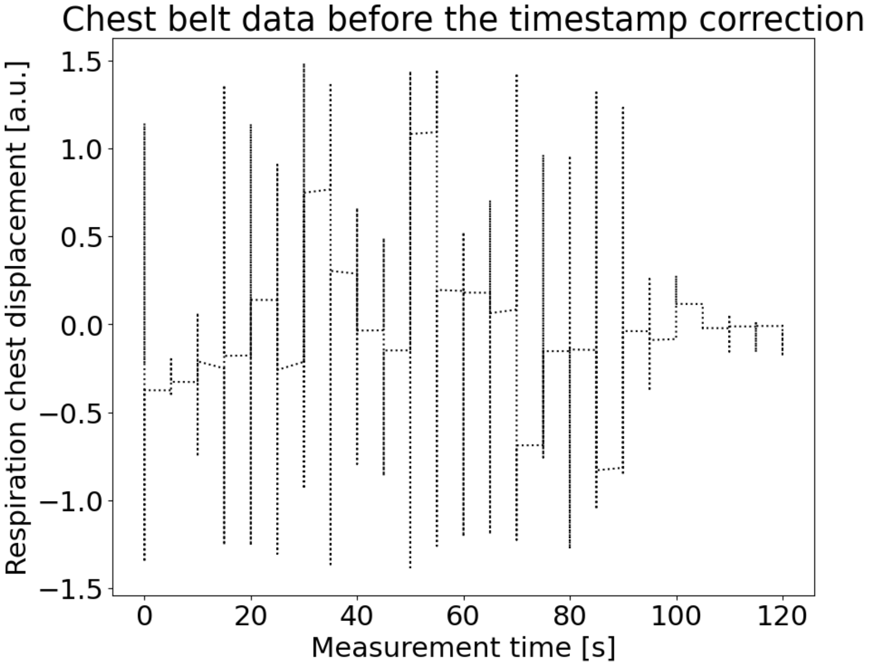


Figure 6.3: Example of chest belt raw data series, before the timestamp correction. The spikes at every 5 s time are caused by a feature in raw data, when 1250 subsequent data points have the same timestamp string. We addressed this feature by introducing an incremental 4 ms step to timestamps.

In total, we have used the chest belt to perform additional calibration measurement of chest wall displacement in 53 periods with Measurement IDs in range [`0047`, `0115`]. The raw data after the applied timestamp correction for these measurements is shown in Fig. 6.5 and Fig. 6.6, where we show measurements that lasted the whole period of 1 minute (in fact, (65 ± 1) s. We will use these measurements for peak detection later on.

In Fig. 6.4 we show measurements that we excluded from further processing, because they lasted significantly less than 1 minute: we have excluded measurements `0057` and `0104` as they lasted 18 s and 25s.

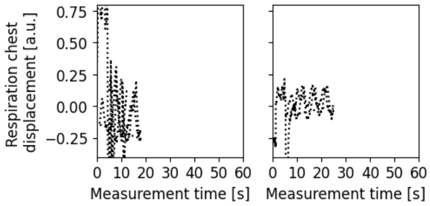


Figure 6.4: Chest belt raw data, excluded data series.

In Fig. 6.5 and Fig. 6.6 the amplitude and its pattern varies wildly, nevertheless, it overall follows a malformed sinusoid-like pattern. We will use this feature to detect respiratory rate from the chest belt data.

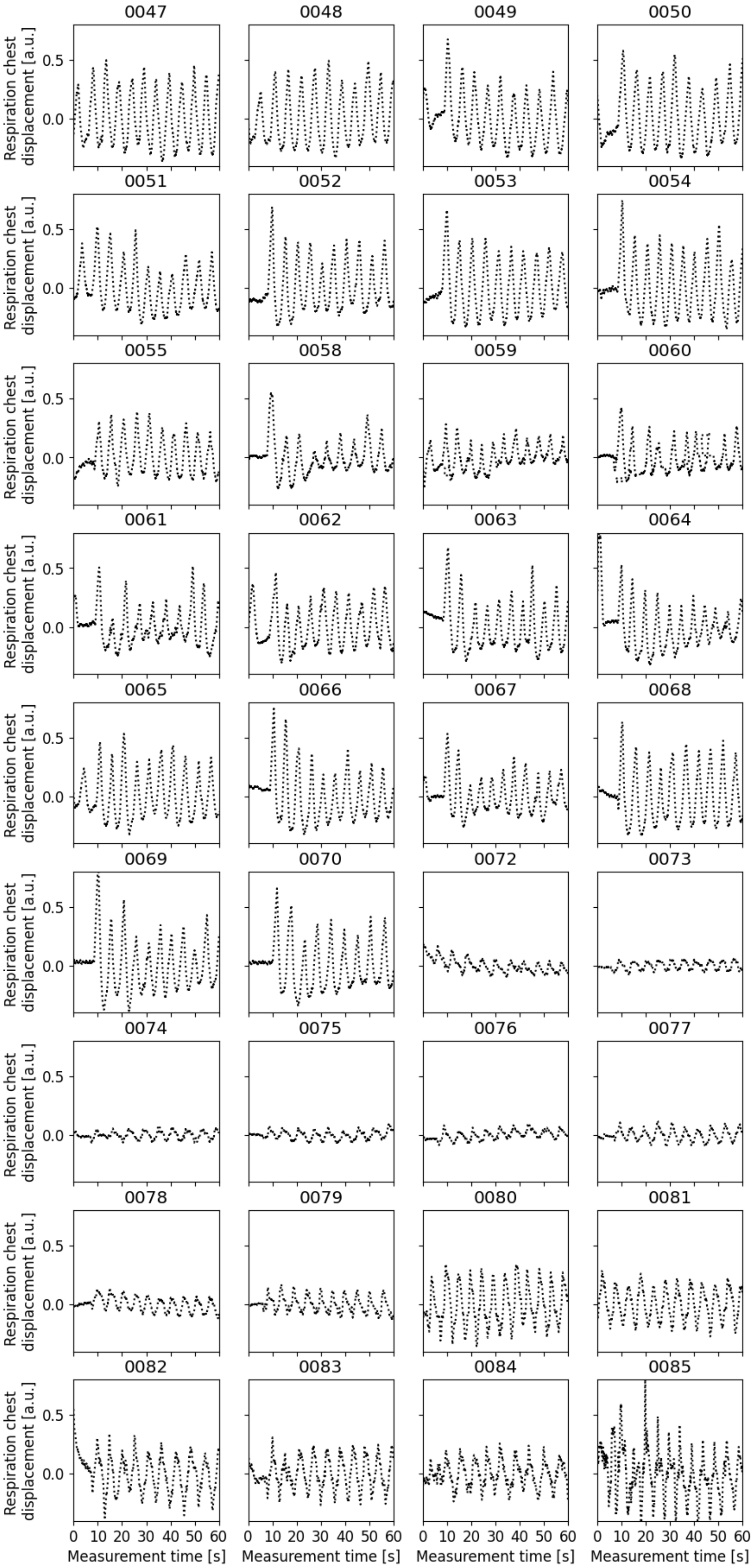


Figure 6.5: Chest belt data after timestamp correction, part 1.

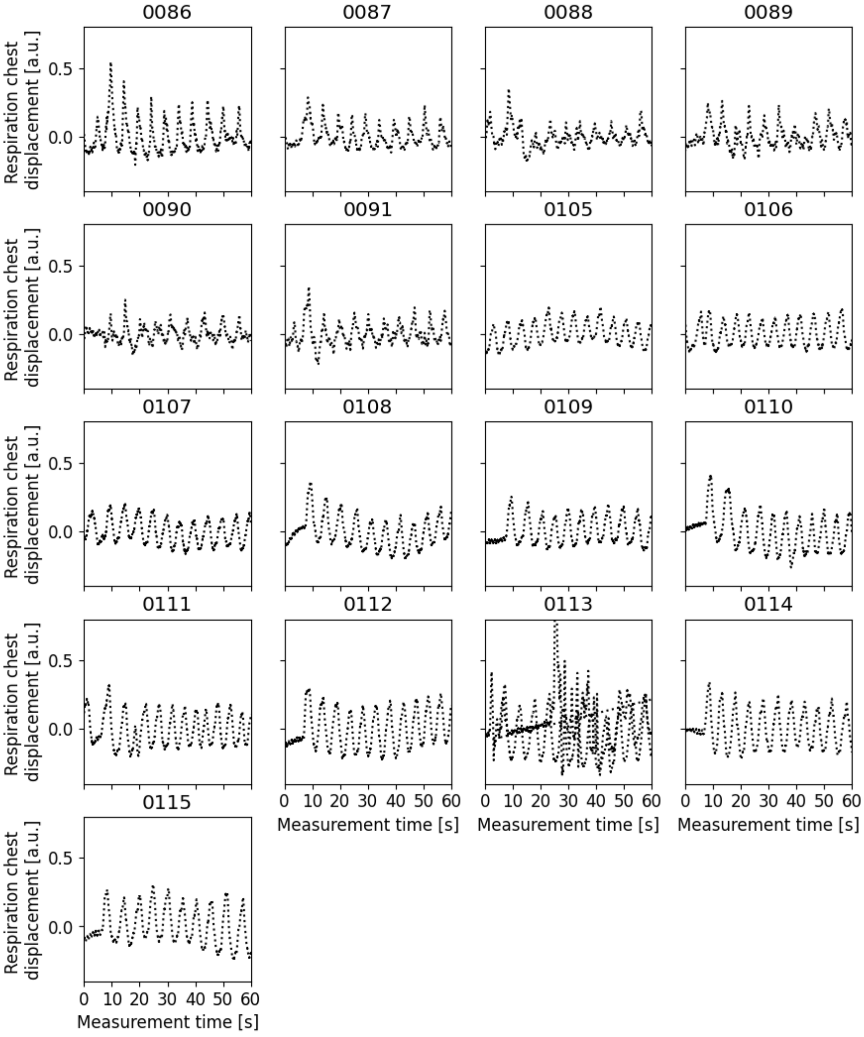


Figure 6.6: Chest belt data after timestamp correction, part 2.

6.2 Peak detection in chest belt data

We need to be able to detect peaks in the data series in order to determine the respiratory rate from the chest belt data. The chest belt data (after the timestamp correction discussed in the previous Section) depicting chest wall displacement during the respiration activity for the relevant data sample is shown in Fig. 6.5 and Fig. 6.6, to demonstrate the peak detection activity we will focus on Measurement IDs [`0047`, `0068`, `0076`, `0085`], shown in Fig. 6.7.

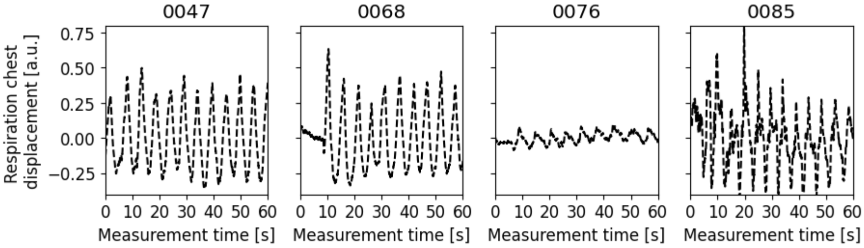


Figure 6.7: Chest belt raw data, sample for peak detection.

In some cases, the sinusoidal shape is very clear (e.g. `0047`), sometimes there is a ”plateau” instead of peak (e.g. first 10 s of `0068`, `0076`), sometimes the smoothed sinusoidal shape is not so clean and contains ”hairy”/spiky peak clusters (e.g. `0076`, `0085`) which then confuse the peak detection algorithms.

We tried three approaches to peak detection:

- manual counting,
- smoothing data, moving average & local peak detection, and
- detecting peaks that are ”enough apart” with `scipy`’s [38] `find_peaks` method.

Manual counting. The manual counting method determines `Ncounted`, and does not scale in terms of automation, however, for the 53 measurements we performed a best-guess-and-count estimation method. For some difficult cases, e.g. for ”plateaus”, we employed both counting and frequency analysis, taking into account the ”plateau” width and frequency of occurences of other peaks. We estimate the manual counting method may have an error of up to 2 breaths on the interval length of (65 ± 1) s.

Code snippets for the two following programmatic peak detection methods are listed in Section A.2.2.

Smoothing data, moving average & local peak detection. With this approach we start with smoothing the data with either 1D Gaussian smoothing, or with a Savitzky-Golay filter [39]. Then we calculate moving averages of the data with a sliding window of length `AVG_WINDOW` (or `window size` in Figures 6.8 through 6.11.), and complete the process by detecting the local maxima positions. We label the number of peaks identified with this method as `Npeaks_gauss` or `Npeaks_savgol`, depending on which smoothing method was chosen.

Detecting peaks "enough apart" with `scipy`'s `find_peaks` method. With this approach we start with identification of a band of the chest wall displacement, where we expect the peaks (of local maxima) to be. We then use the `scipy`'s `find_peaks` method to identify peaks that are at least `DISTANCE_WINDOW` (or `window size`) data points apart. We label the number of peaks identified with this method as `NSciPeaks`.

We studied 42 values of window sizes in a range [100, 900] to understand what would be the best parametric model for peak detection in chest belt data.

In Fig. 6.8 through 6.11 we show how peak detection works for window sizes {100, 420, 460, 520}. In each of these four plots we list the RR determined by the three detection methods (while we do distinguish smoothing in the relevant method), listed in the legend.

Starting with the models parametrized by window size 100 (Fig. 6.8), we found out that for a very smooth sinusoidal shape (`0047`) the peak detection of `Npeaks` and `NSciPeaks` works reasonably well, but with "plateau"-ed and volatile shapes the two methods can differ significantly from the counted RR, or mis-identify a peak in a wrong place (in a minimum point, at the inflection point, or at a local max that is much smaller than a higher local max nearby).

For the three programmatic approaches (smoothing with `gauss`, smoothing with `savgol`, `scipy`), we are trying to understand which window size leads to best matching the peaks position and number of counted peaks. For smoothing with `gauss` and `scipy` the most suitable window size for number of peaks seems to be 520, while for smoothing with `savgol` the most suitable window size seems to be 420.

We varied the window size parameter to observe behavior of the three approaches, e.g. to 420 shown in Fig. 6.9. The detected numbers of peaks are getting closer to the counted ones, however, we can see that e.g. the smoothing approaches systematically lead to avoiding a peak placed on top of the last respiration peak while `scipy` approach does not, and can also see an occurence where `scipy` misses some of the respiration peaks. In terms of peak placement, the "hairy"/spiky measurement (`0085`) shows some peaks identified rather in valleys, although the peak count is close to the counted number of peaks.

In case of window size of 460 (Fig. 6.10), we still see some mis-tagged (valleys) or under-tagged peaks or issue with a "plateau" peak detection, however, for the sample of 4 measurements we may have the closest match of the sample with respect to counted RR, location and frequency of detected peaks.

In case of window size of 520 (Fig. 6.11) we are observing similar features as for window size of 420.

6.2.1 Chest belt peak detection results

We evaluated which method and with what parameters is better for peak detection on the data sample of 53 measurements by calculating average Manhattan distance of `Npeaks` (for two different smoothing methods, `gauss` and `savgol`), and `NSciPeaks` from the `Ncounted`.

As it is clear from the code snippets in Section A.2.2, the smoothing & local max detection approach (`Npeaks`) and `scipy` (`NSciPeaks`) approaches are

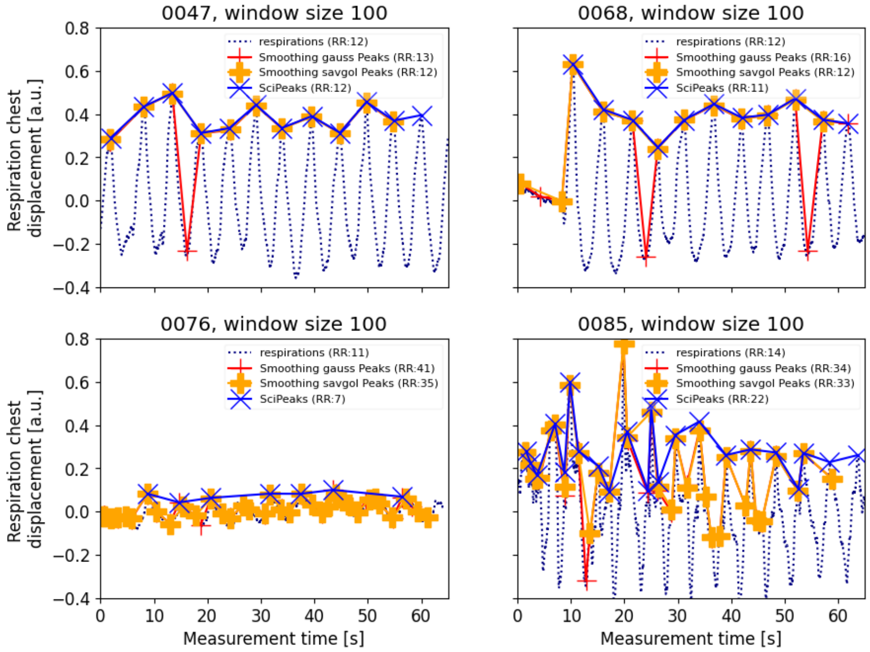


Figure 6.8: Chest belt data peak detection: `window_size = 100`.

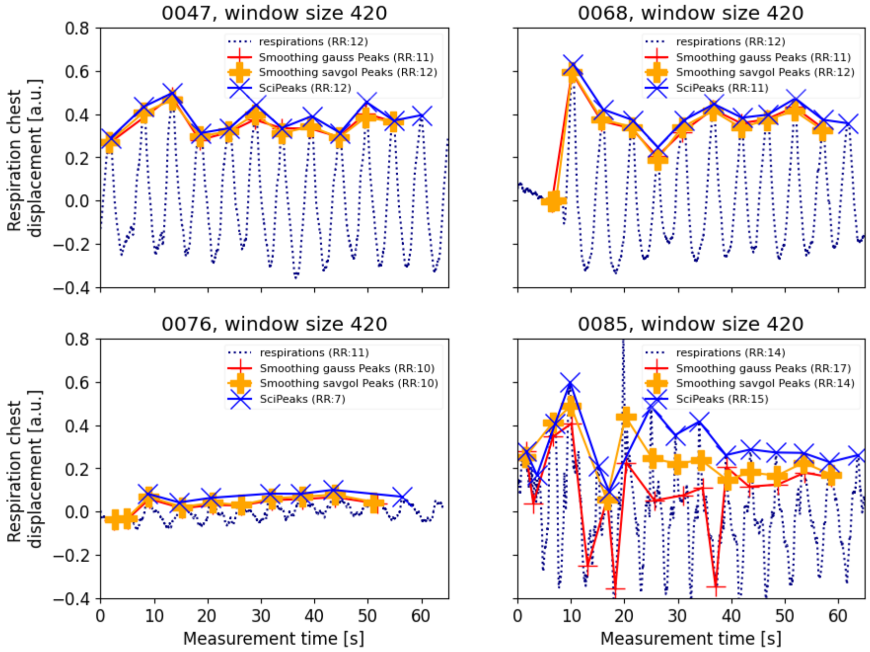


Figure 6.9: Chest belt data peak detection: `window_size = 420`.

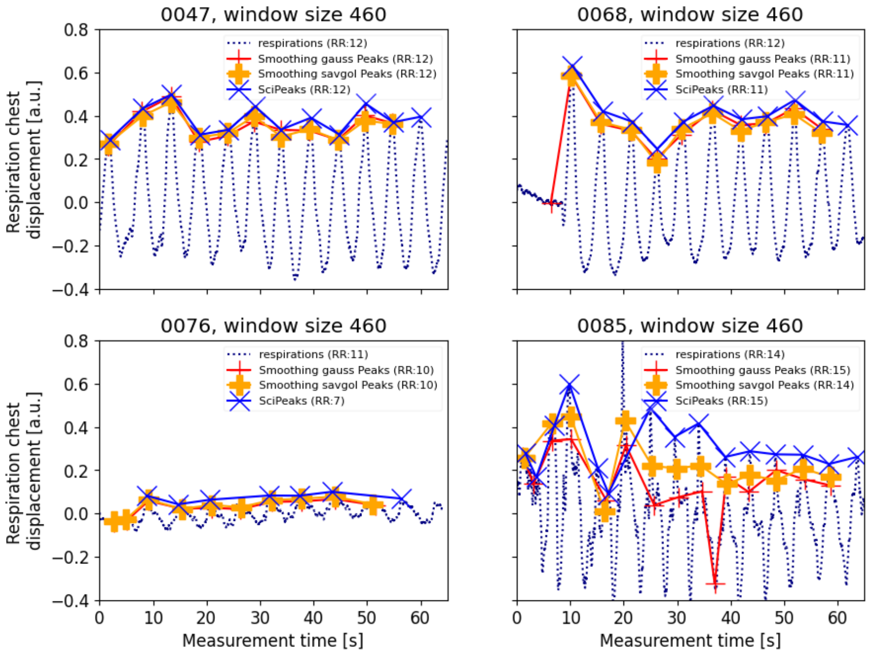


Figure 6.10: Chest belt data peak detection: `window_size = 460`.

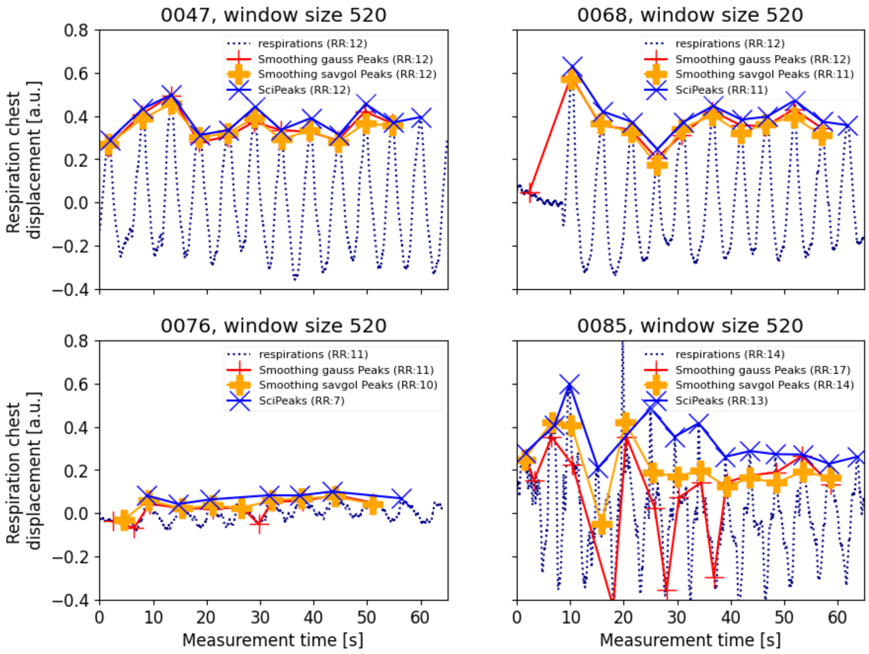


Figure 6.11: Chest belt data peak detection: `window_size = 520`.

independent from each other. To compare them, we use a metric of average Manhattan distance of detected number of peaks from the counted number of peaks for the whole sample of 53 measurements.

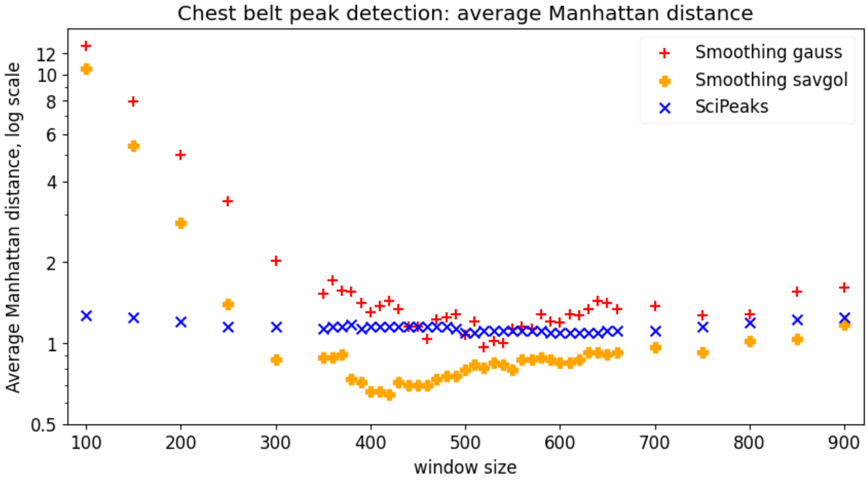


Figure 6.12: Chest belt peak detection, overview of average Manhattan distance by window size.

The comparison of the approaches with different parametric models is shown in Fig. 6.12. This figure considers only detected number of peaks w.r.t. the counted number of peaks, it does not take into account the peak location on top of the respiration peak. We consider a better model one that has the average Manhattan distance closer to zero (keeping in mind that a metric that focuses only on number of peaks may hide mis-, under- or over-tagging). For window sizes outside the range [350, 650] the average Manhattan distance tends to increase significantly, sometimes even tends to explode (the vertical axis has a logarithmic scale).

In Fig. 6.13 we close up on a few examples of window sizes. In this figure we compare `Ncounted`, `Npeaks` and `NSciPeaks` for each measurement. The grayed area denotes a range of (`Ncounted` ± 3).

The smoothing `savgol` model depends on parameters `smoothing` & `window size`. In general, it identifies number of peaks with systematically lower average Manhattan distance of `Npeaks_savgol` from `Ncounted` for the full sample of 53 measurements than both the smoothing `gauss` and the `scipy` approaches, therefore it may be the most suitable model for peak detection, provided that a sensible choice of the window size is made.

The smoothing `gauss` model . In general, it identifies number of peaks with systematically highest average Manhattan distance of `Npeaks_gauss` from `Ncounted` for the full sample of 53 measurements than both the smoothing `savgol` and the `scipy` approaches, and systematically tends to overestimate number of peaks compared to counted number of peaks.

The `NSciPeaks` model depends on parameter `window size`. With the chosen window size values, this method detect number of peaks systematically on the lower side of number of detected peaks, and lays on the middle ground in terms of average Manhattan distance of `NSciPeaks` from `Ncounted` for the full sample of 53 measurements.

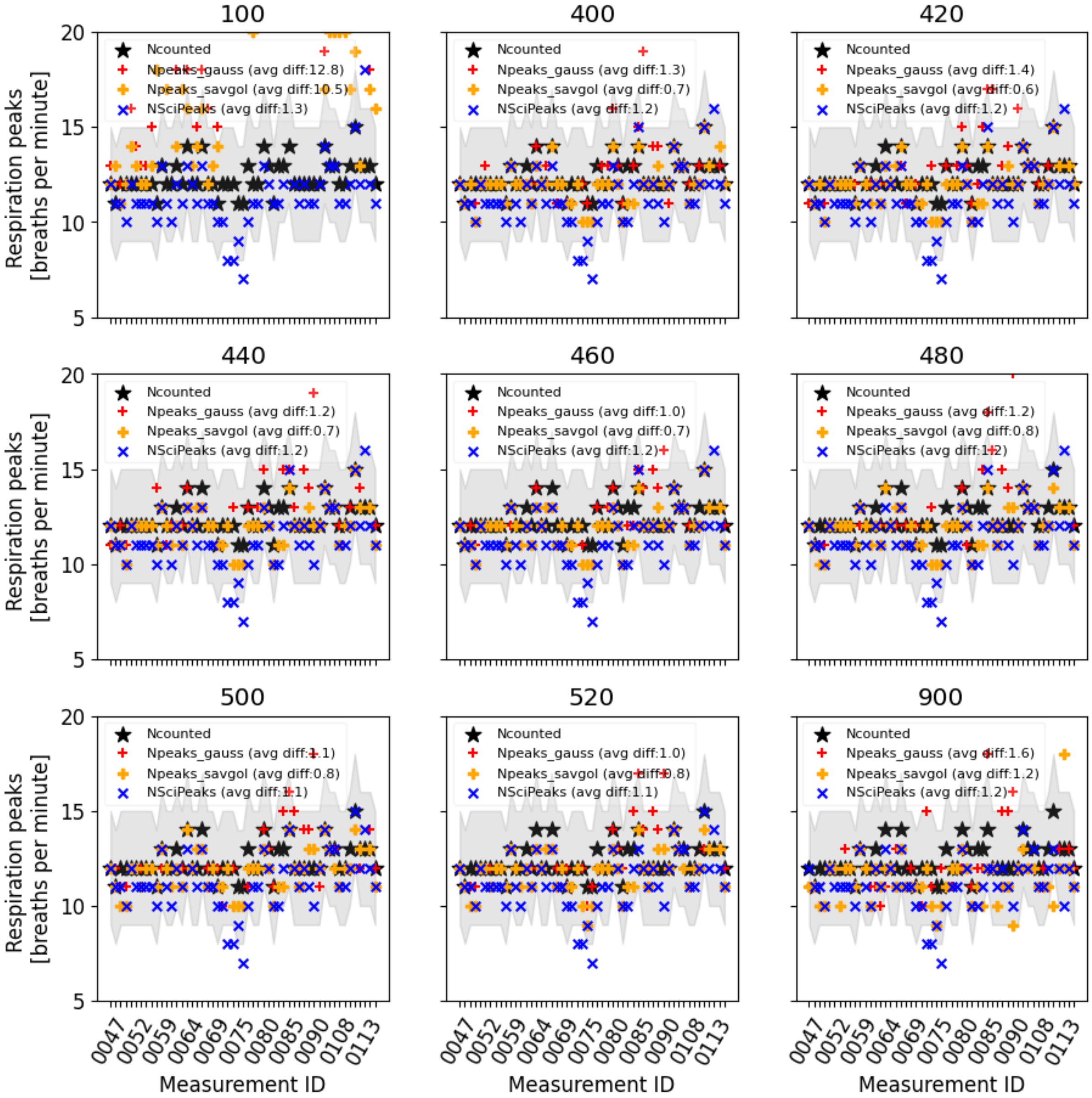


Figure 6.13: Chest belt peak detection, several examples of Manhattan distance distribution for a sample of window size parameters.

7. Distance laser signal processing

In this chapter we describe signal processing from the Baumer OM70 distance laser sensor (Section 3.2). First, we describe data acquisition in Section 7.1, and the respiratory rate detection approach in Section 7.2.

7.1 Data acquisition with Baumer OM70 distance laser

We used the Baumer laser as an independent contact-less calibration sensor in addition to RR detection by the radar. During the data taking period, the laser was attached to the radar with a 3D printed plastic mount, as shown in Fig. 7.1.

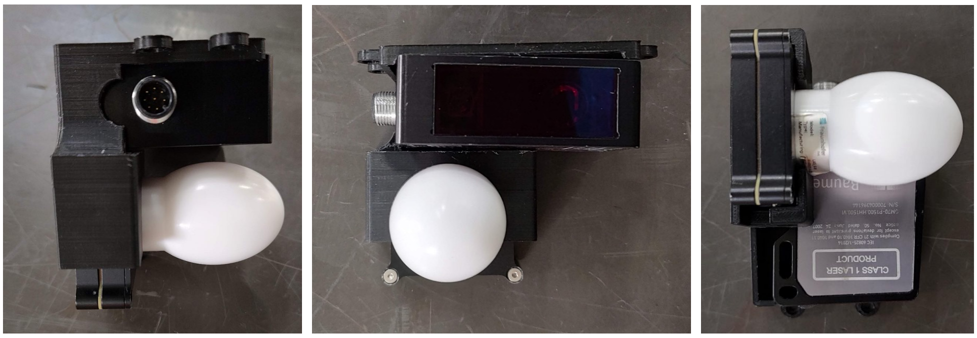


Figure 7.1: Experimental setup of Baumer laser and Fraunhofer radar.

The distance laser was connected to a computer through a USB port, which served for serial communication, while it was powered by a battery pack. We wrote a Python 3 script (code snippet is provided in Section A.4) to configure the distance laser and read out data from it, leveraging the PySerial [23] library.

```
...
X:29:09.742013 : NaN  [b’:01A;NaN;11;454E\r\n’]
X:30:26.186885 : 1338.519  [b’:01A;1338.519;13;1A34\r\n’]
X:30:26.299012 : 1368.027  [b’:01A;1368.027;13;F818\r\n’]
X:30:26.410834 : 1373.601  [b’:01A;1373.601;13;EFD5\r\n’]
...
```

Figure 7.2: Example of a laser data series log.

The used laser measures distances in the range [0.15 m, 1.5 m]. Outside this range it provides only a `NaN` values. The laser data example is shown in Fig. 7.2. The data line contains a timestamp, measurement value (either a distance in mm, or a NaN for an out-of-range measurement), and a bytestring that we read out from the laser device. This bytestring is a semi-colon (’`;`’) separated list of four values, where the second value is the distance we are looking for. A measurement visualisation is shown in Fig. 7.3.

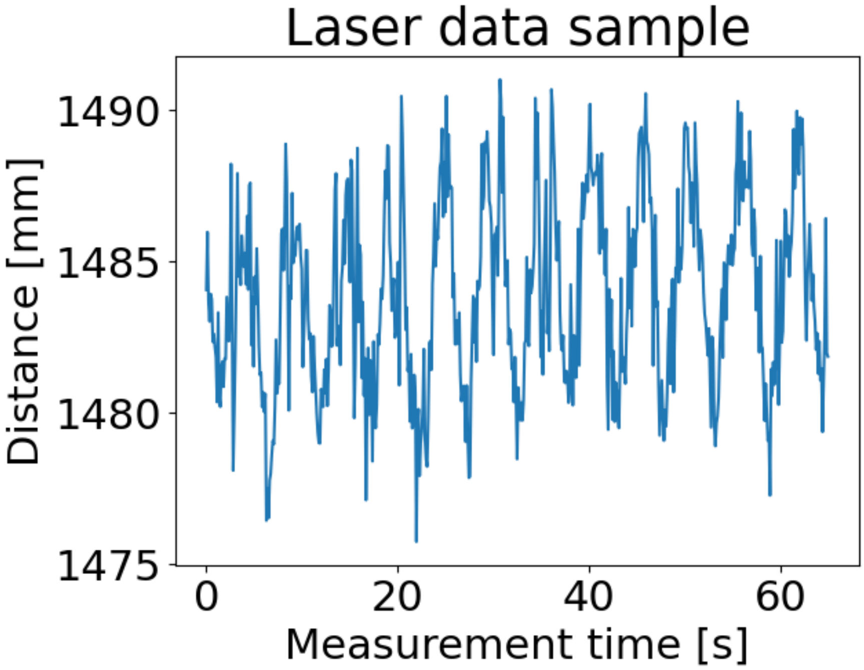


Figure 7.3: Example of a distance laser measurement.

The distance laser data acquisition ran through the whole session with multiple radar measurements, thus we used procedure described in Section 5.6 to identify the periods when respiration rate was detected by radar.

In total, we have used the distance laser to perform additional calibration measurement of chest wall displacement in 58 periods with Measurement IDs in range [`0047`, `0115`].

7.2 Peak detection in distance laser data

We determine the respiratory rate from the distance laser data through detecting peaks. The distance laser data depicting chest wall displacement during the respiration activity for the relevant data sample is shown in Fig. 7.5 and Fig. 7.6, to demonstrate the peak detection activity we will focus on Measurement IDs [`0047`, `0050`, `0075`, `0079`], shown in Fig. 7.4.

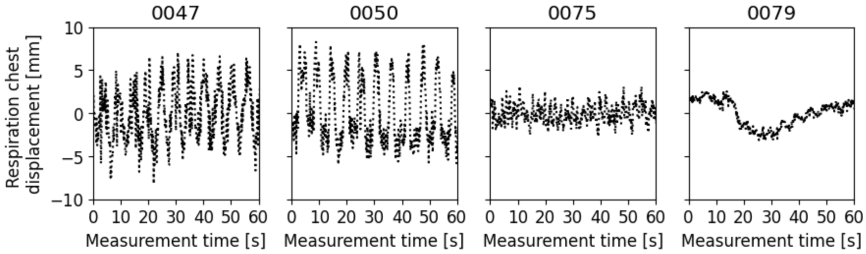


Figure 7.4: Distance laser data, sample for peak detection.

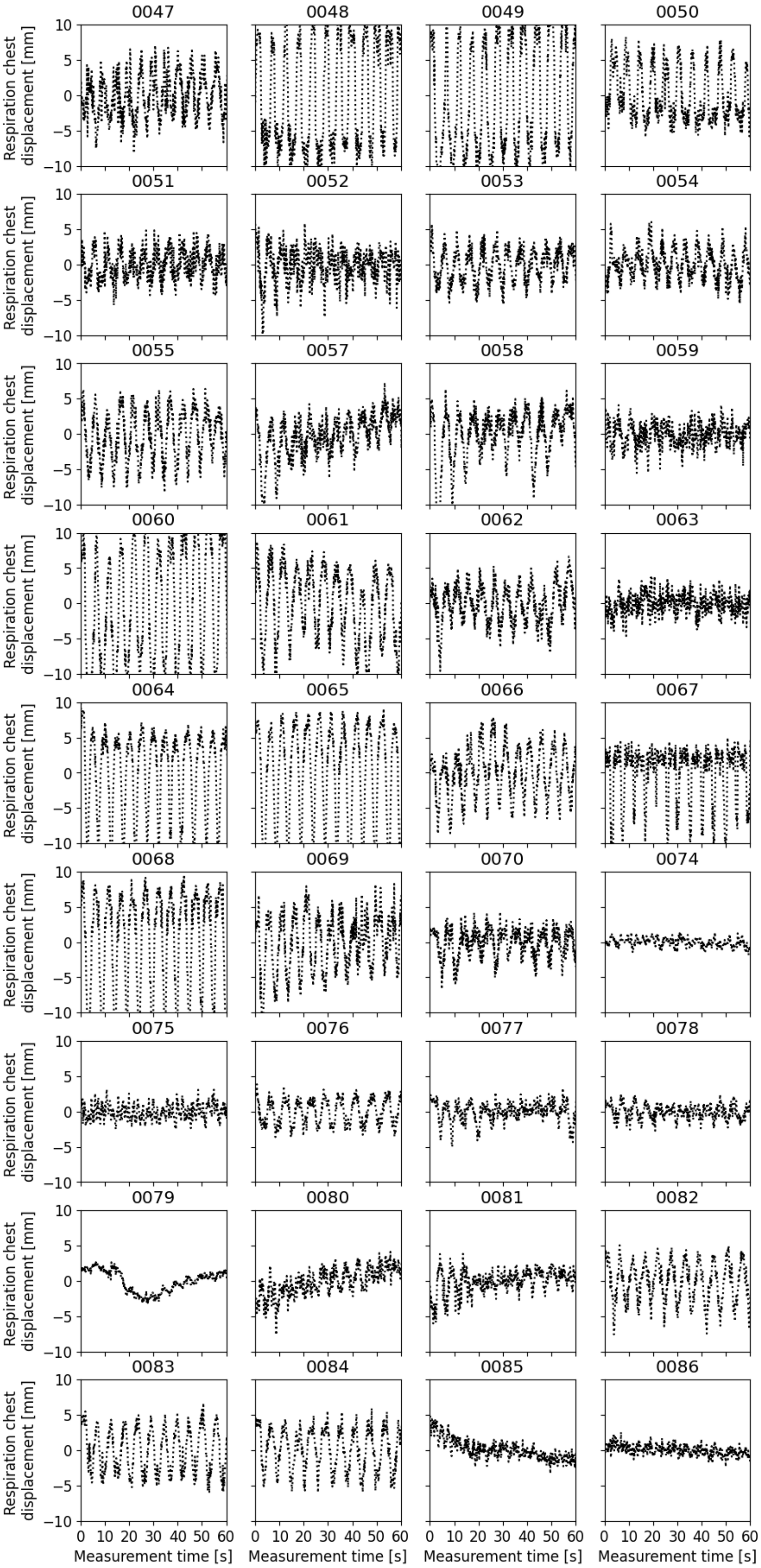


Figure 7.5: Distance laser data, part 1.

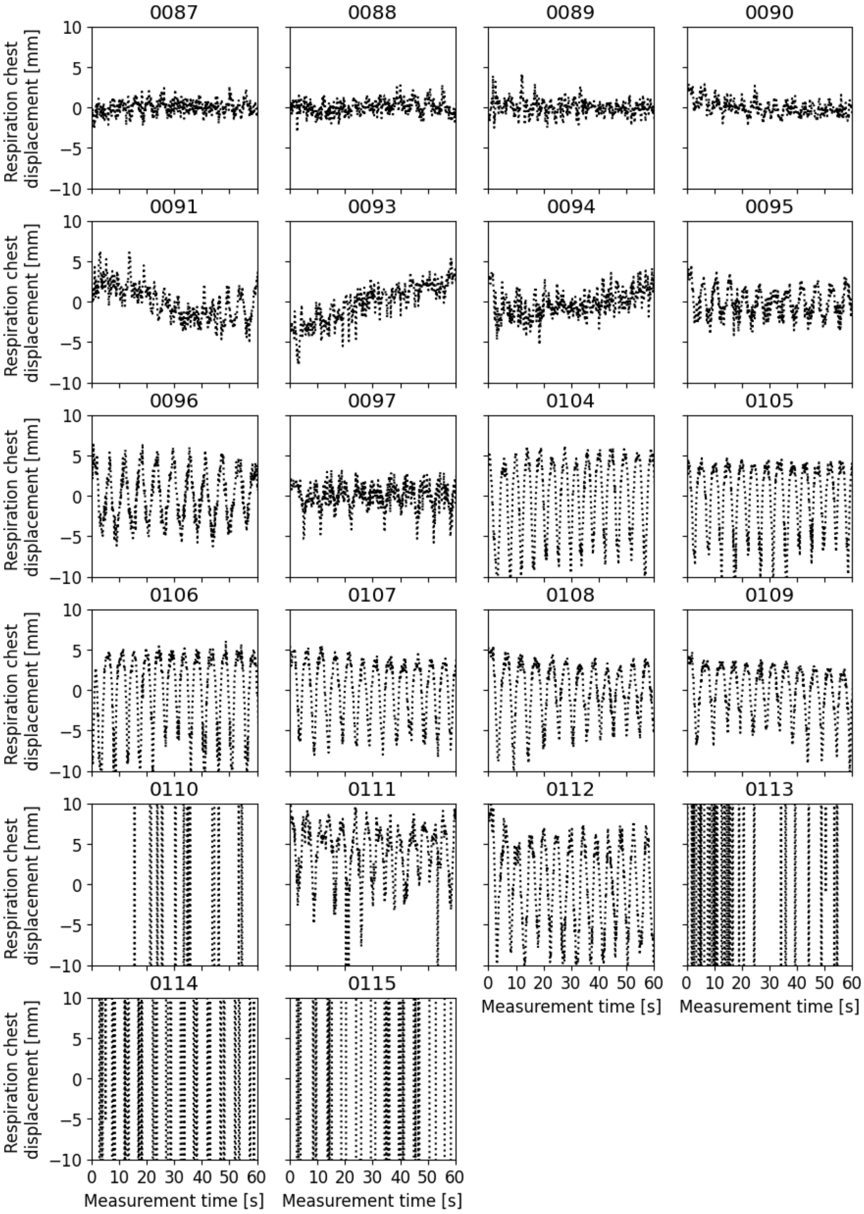


Figure 7.6: Distance laser data, part 2.

In some cases, the sinusoidal shape is very clear (e.g. `0050`), sometimes the smoothed sinusoidal shape is not so clean and contains ”hairy”/spiky peak clusters (e.g. `0047`, `0075`), and sometimes there are even several convoluted sinusoidal shapes (e.g. `0079`) which confuse the peak detection approaches.

We tried the same three approaches to peak detection as in the case of the chest belt:

- manual counting,
- smoothing data, moving average & local peak detection, and
- detecting peaks that are ”enough apart” with `scipy`’s `find_peaks` method.

The difference is in the model parametrizations: for peak detection in the distance laser signal we used different `window size` range, and different parametrization of the `scipy`’s `find_peaks` method – the peak detection code snippet is shown in Section A.2.3.

We studied 32 values of window sizes in a range [10, 50] to understand what would be the best parametric model for peak detection in chest belt data.

In Fig. 7.7 through 7.10 we show how peak detection works for window sizes {10, 20, 35, 50 }. In each of these four plots we list the RR determined by the three detection methods (while we do distinguish smoothing in the relevant method), listed in the legend.

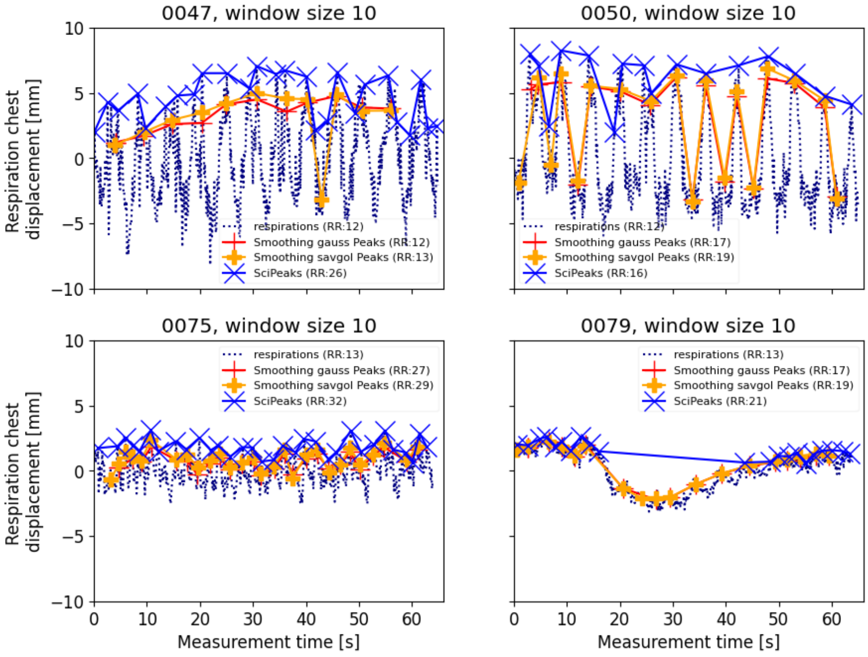


Figure 7.7: Laser data peak detection: `window_size = 10`.

Starting with the models parametrized by `window_size = 10` (Fig. 7.7), we see that for the first three data series the three approaches systematically over-tag the peaks, locating estimated peaks in both peak and valley positions. In the ”convoluted sinusoids” measurement `0079`, the smoothing methods over-tag the

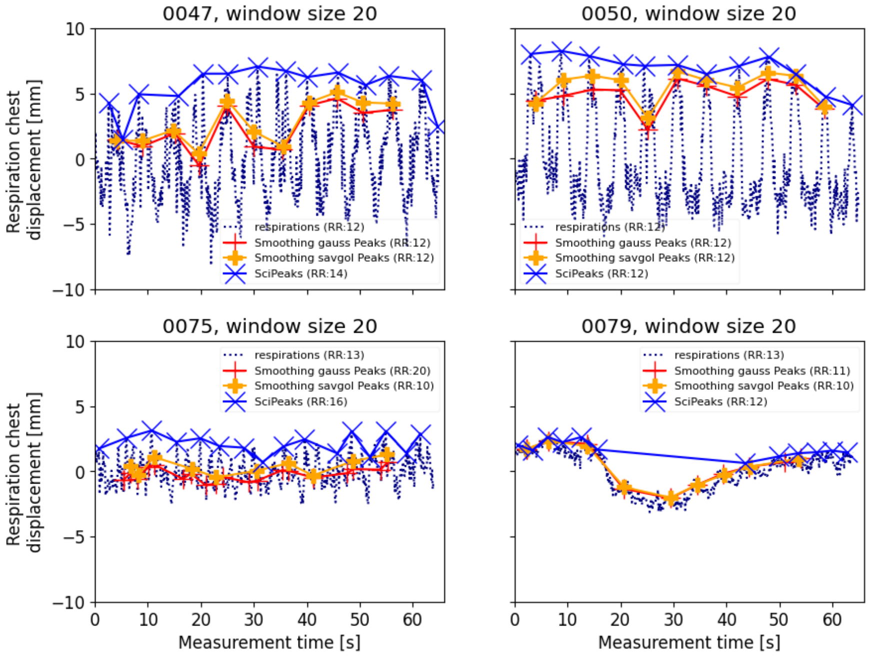


Figure 7.8: Laser data peak detection: `window_size = 20`.

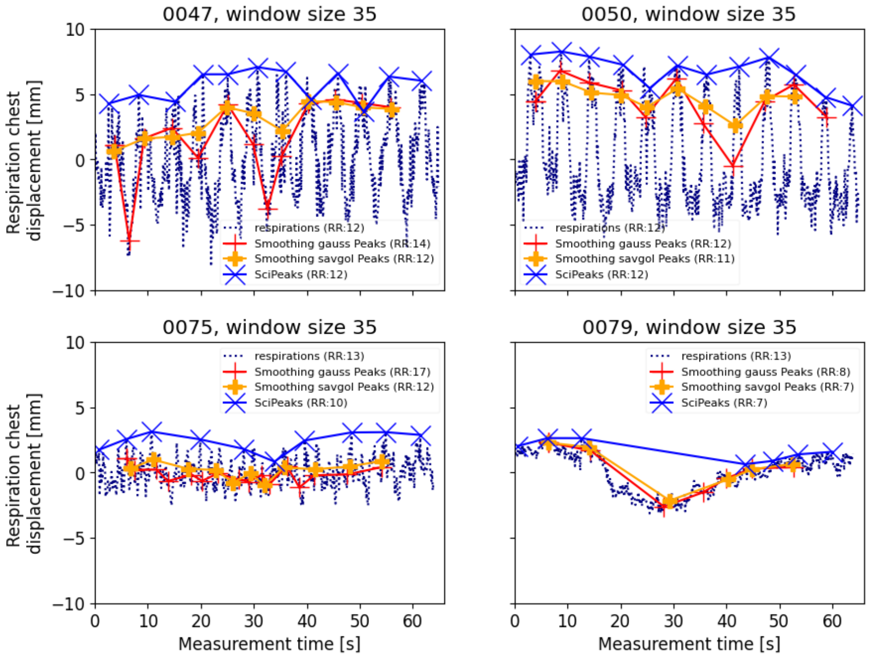


Figure 7.9: Laser data peak detection: `window_size = 35`.

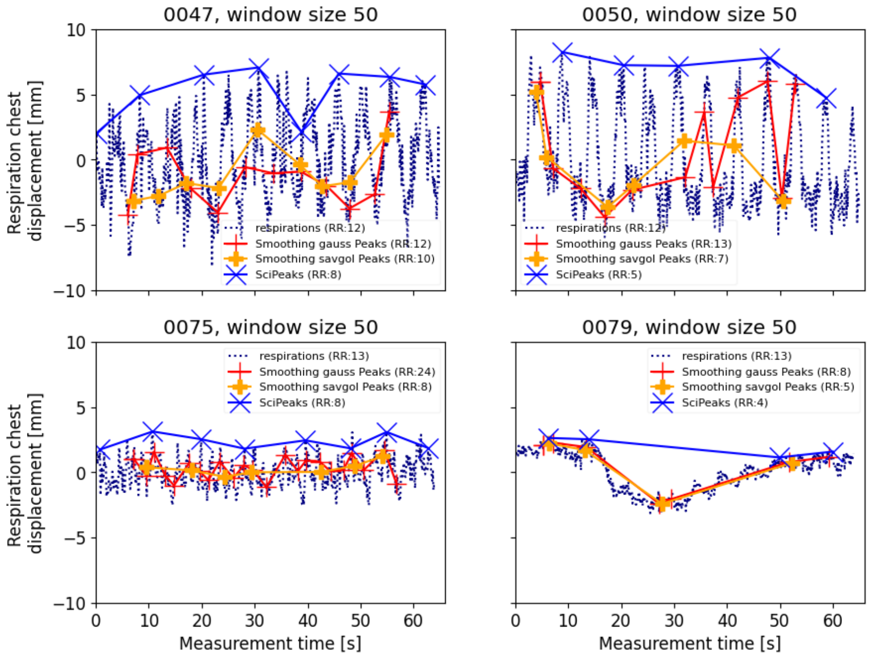


Figure 7.10: Laser data peak detection: `window_size` = 50.

peaks, while the scipeak method has trouble detecting peaks in the large valley (time [20, 40] s).

The `window_size = 20` parametrization (Fig. 7.8) represents a ”sweet spot” where the three approaches show the closest peak count w.r.t. counted number of peaks. For `window_size = 35` (Fig. 7.9) and `window_size = 50` (Fig. 7.10) again show detected peak counts diverging away from the counted one.

7.2.1 Distance laser peak detection results

We evaluated which method and with what parametrization is better for peak detection on the data sample of 58 measurements by calculating average Manhattan distance of `Npeaks` (for two different smoothing methods, `gauss` and `savgol`), and `NSciPeaks` from the `Ncounted`.

The code snippets in Section A.2.3 shows that the smoothing & local max detection approach (`Npeaks`) and `scipy` (`NSciPeaks`) approaches are independent from each other. We compare them, we use again a metric of average Manhattan distance of detected number of peaks from the counted number of peaks for the whole sample of 58 measurements.

The comparison of the approaches with different parametric models is shown in Fig. 7.11. This figure considers the detected number of peaks w.r.t. the counted number of peaks, and leaves the peak location on top of the respiration peak out of consideration. We consider a better model to be one that has the average Manhattan distance closer to zero. For window sizes outside the range [15, 35] the average Manhattan distance tends to increase significantly, sometimes even exploding.

In Fig. 7.12 we show in more details a few examples of window sizes. In this figure we compare `Ncounted`, `Npeaks` and `NSciPeaks` for each measurement. The grayed area denotes a range of `Ncounted` ± 3.

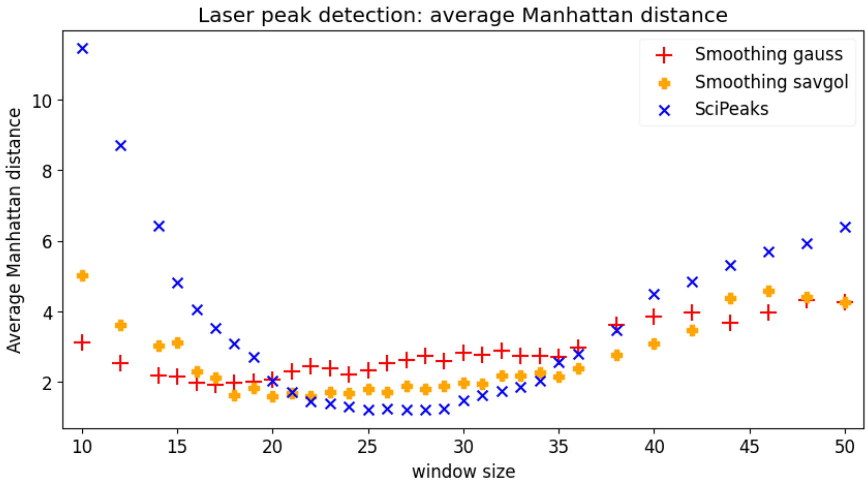


Figure 7.11: Laser peak detection, overview of average Manhattan distance by window size.

The `NSciPeaks` model depends on parameter `window size`. With the studied `window size` values [10, 50], this method detect number of peaks systematically lowest average Manhattan distance over the full sample of 58 measurements, however also systematically under-tags ”hairy”/spiky data series. The best performance (with lowest average Manhattan distance, and the detected peak count within the range (`Ncounted` ± 3) is observed for `window size` $\in [20, 33]$. For `window size` < 20 this approach over-tags the peaks, and for `window size` > 33 it under-tags the peaks.

The smoothing `savgol` model depends on parameters `smoothing` & `window size`. In general, it identifies number of peaks with systematically the middle-ground value of average Manhattan distance of `Npeaks_savgol` from `Ncounted` for the full sample of 53 measurements than both the smoothing `gauss` and the `scipy` approaches, therefore also here it may be the most suitable model for peak detection, provided that a sensible choice of the `window size` is made – for `window size` $\in [20, 33]$.

The smoothing `gauss` model: In general, it identifies number of peaks with systematically highest average Manhattan distance of `Npeaks_gauss` from `Ncounted` for the full sample of 53 measurements than both the smoothing `savgol` and the `scipy` approaches, and systematically tends to overestimate number of peaks compared to counted number of peaks, also in the range of `window size` $\in [20, 33]$. Therefore, this approach looks like a least fortunate choice for peak detection approach in smoothed distance laser data series, possibly because the data series are very volatile for the smoothing model & local maxima detection, compared to the chest belt data series.

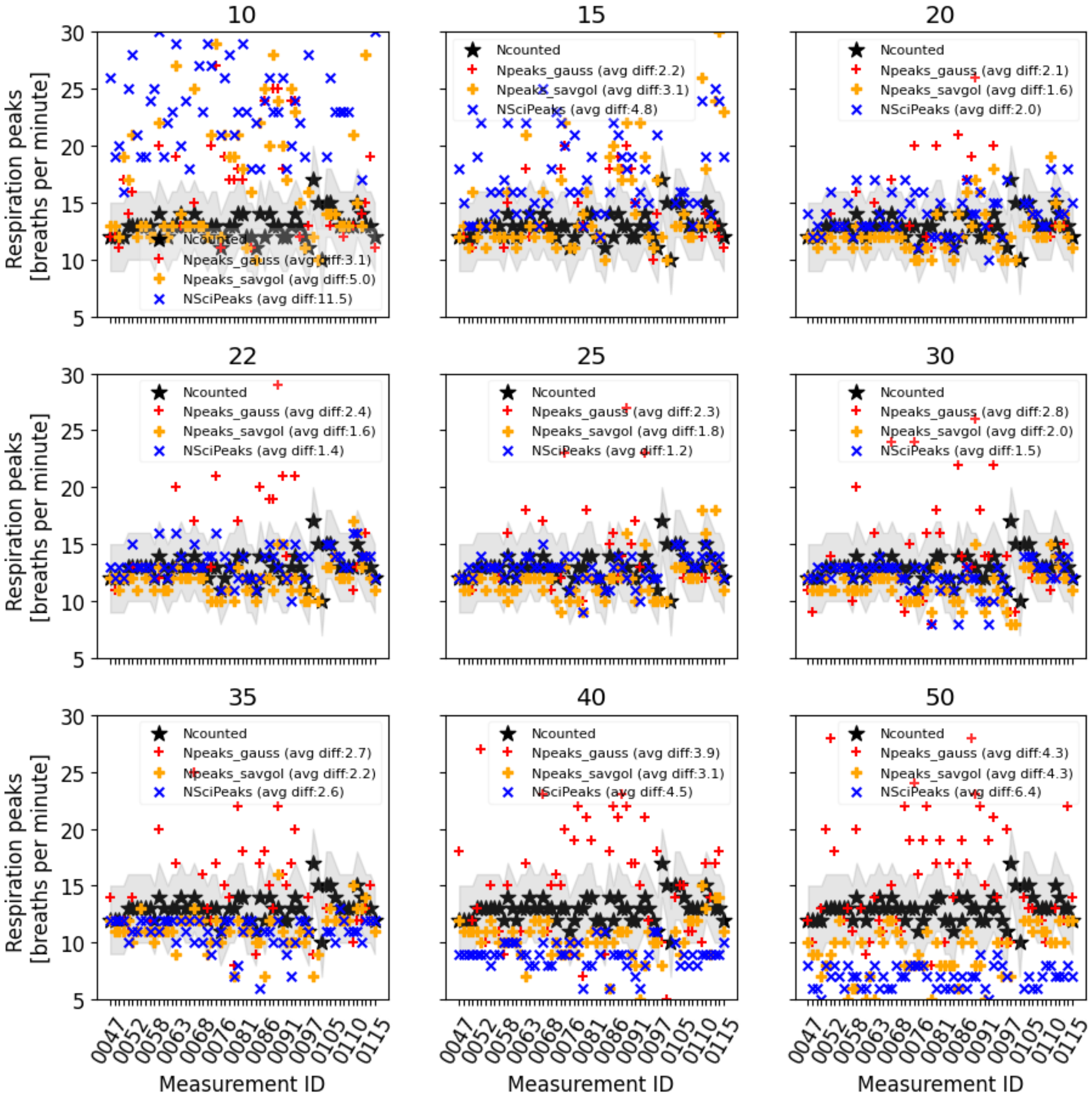


Figure 7.12: Laser peak detection, several examples of Manhattan distance distribution for a sample of window size parameters.

8. Results discussion

In this Chapter we summarize our studies of remote detection of human vital signs, particularly respiratory activity, with a FMCW radar, a distance laser, and a contact-full detection with a chest belt.

8.1 Summary: radar data acquisition

At first, we gained an initial understanding of radar data acquisition and its signal processing, as discussed in Chapter 4. In Section 3.4 we proposed 6 measurements scenarios, `M01` through `M06`, and performed these measurements. Bellow we summarize feasibility or challenges encountered:

`M01`: 1D angular resolution spectrum. (35 measurements) In the beginning several attempts to measure the 1D angular resolution spectrum of the radar have been made (in horizontal and vertical axis), however, resulting into a conclusion that with the available laboratory setup, one cannot achieve a precise spectrum measurement with resolution of several degrees.

`M02`: Subject lying down on their back. (8 measurements) In this scenario, detection from different parts of the subject's body has been performed, with the subject lying down on the back, wearing the chest belt. The radar & laser were at a fixed vertical height position, measuring while pointing radar at different parts of the subject's body, e.g. head, shoulders, several chest locations, and hips and thighs. The distance of the radar & laser from the subject was fixed, in the range of 1.1 m to 1.5 m. With these scenarios, a confirmation has been made that the only body area where one can reasonably detect respiratory movements remotely with a radar is the human subject's thorax, therefore further measurements were restricted only to thorax.

`M03`: Subject lying down on their side. (12 measurements) In this scenario, detection of respiratory movements of a human subject lying down on their side has been performed, either facing the radar, or the radar pointed at their back. While facing the radar, the measurements focused at the subject's collar bone, several locations on their chest, waist and hips. While detecting from the person's back, the measurements focused to the person's upper shoulders, scapulas, several places on the chest around their diaphragm, and waist and hips.

`M04`: Subject lying down on their stomach. (5 measurements) In this scenario, respiratory movements of a human subject lying down on their stomach, turned with their side to the radar were detected. The radar was pointing, from the person's side, to locations on their shoulders, scapulas, rib cage line, waistline, and hips. In these scenarios, the human subject was wearing a t-shirt or a light, tightly fitting, sweater.

`M05`: Subject sitting down, wearing a standard-issue work jacket. (5 measurements) In this scenario, the human subject was sitting down in a chair, wearing a standard-issue work jacket. Several measurements, pointing the radar & laser to the area of diaphragm, have been performed. In these measurements there was no obstacle in the line of beam between the radar and the human subject, except for the Earth atmosphere.

`M06`: Subject sitting down, wearing a standard-issue work jacket, with a plexi-glass in the line of beam. (12 measurements) In this scenario, the setup was similar to M05, however, a decision has been made to add a clear plexi-glass obstacle to the line of beam. That turned out to be a killer change parameter, as both the laser and radar data analysis packages heavily suffered from the newly introduced refraction effects.

Besides measuring through these 6 scenarios, we confirmed it is not trivial to bring a new/novel sensor from conception through commissioning to production, and that the engineering behind the whole process can be truly interesting experience. During our measurement sessions we faced issues with the radar overheating, as discussed in Section 4.2.1.

The most suitable conditions for RR detection are a static system with no obstacle in the line of beam, however, we were able to detect RR also during worsened conditions. One of the most important realisations is the need for tightness or fitting of the working clothes: when the work jacket or other work clothing is not loose, that helps to the rescue! That is indeed an important fact for work place safety!

8.2 Summary: radar signal processing

In Chapter 5 we discussed the approaches of Fraunhofer FMCW radar signal processing that we took. We discussed what is the signal processing cut flow, what challenges we encountered and addressed, in software for data acquisition, and for signal processing.

We developed the `health_detection` application, to scan the phase space of parameters of the signal processing parametric models. In Section 5.5 we described parametrizations of two families of parametric models: one based on CFAR algorithm peak detection described in (5.19), and another one which we implemented, based on CWT algorithm peak detection described in (5.20).

8.3 Summary: calibration sensors

We discussed the data acquisition and signal processing of the PLUX chest belt data in Chapter 6, and of the Baumer OM70 distance laser data in Chapter 7.

We described three approaches for RR detection for both sensors: 1) manual counting, 2) smoothing data & local peak detection; with either Gaussian smoothing, or with a Savitzky-Golay filter, and 3) detecting peaks that are ”enough apart” with `scipy`’s [38] `find_peaks` method.

We summarized the peak detection results, for the chest belt signal processing in Section 6.2.1, and for the distance laser signal processing in Section 7.2.1.

9. Future work

In this chapter we briefly describe what could be the possible paths towards improvements of the respiratory rate detection performed and described in this work.

Earlier we described the limitations coming from environmental, sensor device, nature of respiratory activity, and signal processing areas.

In the data acquisition area, one of the improvements could be an introduction of an active cooling mechanism of the radar device, where we hope to allow for un-interrupted data acquisition periods of many hours, maybe even several days, in a row.

Other listed limitations boil down to improvements in signal processing.

A dynamic breathing pattern, a short and shallow breathing or gasping, is difficult to detect, as our parametric model is built around variance cuts. We hope that with a different, and possibly a more complex signal processing / peak detection approach we would be able to overcome the high data volatility and detect RR.

Another interesting area of further research would be studying alternative sensing methods for an environment where we do not have a clear line of sight, when there are obstacles along the line of beam, e.g. a plexi-glass or smoke. In addition to the radar (being mounted on a robotic arm of a mobile robot), one could equip the robot with a IR camera (and use e.g computer vision techniques to detect movements in the field of view). One could equip the robot with a microphone or even a two-way communication device, to allow the remote Fire & Rescue operator to listen or even to communicate with the person(s) in distress.

10. Conclusions

In this work we studied remote sensoric detection of human vital signs with a frequency-modulated continuous wave radar. We achieved our objectives:

Data acquisition: The used Fraunhofer FMCW 80 GHz radar was a prototype sensor, where a proper (possibly proprietary) documentation of its data format was not available. Despite it, we were able to successfully acquire sensoric data from two contact-less sensors (Fraunhofer FMCW 80 GHz radar, and a Baumer OM70 distance laser) and a contact-ful sensor (PLUX chest belt), and developed software where we implemented parametric models to successfully detect respiratory rate in the acquired data.

We described several families of success scenarios of respiratory rate detection in the radar signal.

We implemented several improvements in the radar signal detection: ranging from identifying the best max variance thresholds, through experimentation with the choice of central method for outliers removal, to providing an alternative approach to the peak detection algorithm and identifying its initial parametrization, so that it provides viable information for RR detection.

We implemented the `health_detection` application to study different parametric models to detect respiratory rate in radar signal.

We described limitations of the experimental setup:

- environmental: need of clear line of sight / no obstacles,
- sensor device: need for active cooling,
- nature of human respiratory activity: need to target body area around the person's diaphragm; need for certain breathing patterns: shallow or quick breathing or person movements bring too much volatility to the signal,
- signal processing: limitations of the parametric models on success of RR detection, e.g. high volatility signal processing needs complex de-compensation of signal volatility in terms of variation management and peak detection that goes beyond the scope of this work.

We documented environment dependencies of the `HealthDetectionSample` application, and made it a standalone application. We created a containerized environment to run this application in a container with Podman.

We demonstrated that a FMCW radar is a viable sensor for a remote detection of human respiratory activity, and described several parametric models to increase chances of detecting a reasonably correct respiratory rate, as well as provided an understanding of the limitations of the experimental setup and the parametric models.

Bibliography

[1] Wikipedia contributors. Radar. https://en.wikipedia.org/wiki/Radar.

[2] Sandeep Rao. Introduction to mmwave Sensing: FMCW Radars. https://www.ti.com/video/series/mmwave-training-series.html.

[3] Mamady Kebe, Rida Gadhafi, Baker Mohammad, Mihai Sanduleanu, Hani Saleh, and Mahmoud Al-Qutayri. Human vital signs detection methods and potential using radars: A review. *Sensors*, 20(5), 2020.

[4] Hideo Kaneko and Jun Horie. Breathing movements of the chest and abdominal wall in healthy subjects. *Respiratory Care*, 57(9):1442–1451, 2012.

[5] J M Moll and V Wright. An objective clinical study of chest expansion. *Annals of the Rheumatic Diseases*, 31(1):1–8, 1972.

[6] Gary G. Berntson, John T. Cacioppo, and Karen S. Quigley. Respiratory sinus arrhythmia: Autonomic origins, physiological mechanisms, and psychophysiological implications. *Psychophysiology*, 30(2):183–196, 1993.

[7] Carlo Massaroni, Andrea Nicolò, Daniela Lo Presti, Massimo Sacchetti, Sergio Silvestri, and Emiliano Schena. Contact-based methods for measuring respiratory rate. *Sensors*, 19(4), 2019.

[8] P. Höppe. Temperatures of expired air under varying climatic conditions. *International Journal of Biometeorology*, 25(2):127–132, June 1981.

[9] Carlo Massaroni, Daniela Lo Presti, Paola Saccomandi, Michele Arturo Caponero, Rosaria D'Amato, and Emiliano Schena. Fiber bragg grating probe for relative humidity and respiratory frequency estimation: Assessment during mechanical ventilation. *IEEE Sensors Journal*, 18(5):2125–2130, 2018.

[10] Nils Pohl, Timo Jaeschke, and Klaus Aufinger. An ultra-wideband 80 ghz fmcw radar system using a sige bipolar transceiver chip stabilized by a fractional-n pll synthesizer. *IEEE Transactions on Microwave Theory and Techniques*, 60(3):757–765, 2012.

[11] Sandra Nowok, Gunnar Briese, Simon Kueppers, and Reinhold Herschel. 3d mechanically pivoting radar system using fmcw approach. In *2018 19th International Radar Symposium (IRS)*, pages 1–10, 2018.

[12] Baumer Electric AG. Baumer Distance Sensor OM70. https://www.baumer.com/medias/__secure__/Baumer_OM70-L0140.HH0070.VI_EN_20210409_DS.pdf?mediaPK=9017362186270.

[13] PLUX - Wireless Biosignals, S.A. PLUX Piezo-Electric Respiration (PZT) Sensor. https://support.pluxbiosignals.com/wp-content/uploads/2021/11/Respiration_PZT_Datasheet.pdf.

[14] NASA. Microphones on the Perseverance Rover. https://mars.nasa.gov/mars2020/spacecraft/rover/microphones/.

[15] Pat Brennan. Another First: Perseverance Captures the Sounds of Driving on Mars. https://mars.nasa.gov/news/8892/another-first-perseverance-captures-the-sounds-of-driving-on-mars/.

[16] Andrew Good. Hear Sounds From Mars Captured by NASA's Perseverance Rover. https://www.jpl.nasa.gov/news/hear-sounds-from-mars-captured-by-nasas-perseverance-rover.

[17] Markus Tarin. Thermal Infrared Imaging in IR Cameras - Explained! https://movitherm.com/knowledgebase/thermal-infrared-imaging-explained/.

[18] European Union. General Data Protection Regulation. https://gdpr-info.eu/.

[19] CERN. The processing of personal data at CERN. Traitement des données à caractère personnel au CERN. https://cds.cern.ch/record/2651311, Jan 2019.

[20] CERN. Use of CERN computing facilities. Utilisation des installations informatiques du CERN. https://cds.cern.ch/record/1202776, Oct 2000.

[21] A.S. Fiorillo, C.D. Critello, and S.A. Pullano. Theory, technology and applications of piezoresistive sensors: A review. *Sensors and Actuators A: Physical*, 281:156–175, 2018.

[22] Carlos Veiga Almagro, Mario Di Castro, Giacomo Lunghi, Raúl Marín Prades, Pedro José Sanz Valero, Manuel Ferre Pérez, and Alessandro Masi. Monocular robust depth estimation vision system for robotic tasks interventions in metallic targets. *Sensors*, 19(14), 2019.

[23] pySerial Authors. pySerial. https://pyserial.readthedocs.io/.

[24] Yee Siong Lee, Pubudu N Pathirana, Terry Caelli, and Saiyi Li. Further applications of doppler radar for non-contact respiratory assessment. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3833–3836, 2013.

[25] Olga Boric-Lubecke, Victor M. Lubecke, Isar Mostafanezhad, Byung-Kwon Park, Wansuree Massagram, and Branka Jokanović. Doppler Radar Architectures and Signal Processing for Heart Rate Extraction. In *Mikrotalasna Revija*, volume 15, 2009.

[26] Louis L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley, 1991.

[27] CMake. CMake. https://cmake.org/.

[28] Free Software Foundation. GNU Make. https://www.gnu.org/software/make/.

[29] Free Software Foundation. GNU Compiler Collection. https://gcc.gnu.org/.

[30] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

[31] RedHat. Podman (the POD manager). https://podman.io/.

[32] John Sadowsky. The Continuous Wavelet Transform: A Tool for Signal Investigation and Understanding. *The Johns Hopkins University Technical Digest*, 15(4):306–318, 1994.

[33] The PyWavelets Developers. PyWavelets - Wavelet Transforms in Python. https://pywavelets.readthedocs.io/en/latest/.

[34] The NumPy Developers. NumPy. https://numpy.org/.

[35] OpenAI. ChatGPT (Feb 2023) [Large language model]. https://chat.openai.com.

[36] pyan contributors. pyan3: Offline call graph generator for Python 3. https://pypi.org/project/pyan3/.

[37] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz— open source graph drawing tools. In Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing*, pages 483–484, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[38] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[39] Abraham Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.*, 36(8):1627–1639, July 1964.

[40] Executable Books Community. Jupyter Book. https://doi.org/10.5281/zenodo.4539666.

A. Attachments

In this chapter we provide various code snippets used in processing of signal from radar, chest belt and laser measurements and detection of respiratory rate. Majority of the code snippets are provided in the attachments archive file, smaller part is listed in the following sections.

First, we describe the attachments archive file in Section A.1. There we describe code snippets for intervals endpoint detection in subsection A.2.1, peak detection in chest belt data in subsection A.2.2, peak detection in distance laser data in subsection A.2.3, and the `health_detection` application in subsection A.2.4. These code snippets are provided either in a form of an interactive Jupyter [40] notebook, or a Python 3 script with a bash wrapper.

Then we continue with description of the `HealthDetectionSample` application `Dockerfile` in Section A.3, and close with description of distance laser data acquisition script in Section A.4.

A.1 Archive structure

In this section we describe content of the attachments archive of this work. The attachments archive has the following structure:

```
archive/
|-- README.md
|-- data
|   |-- 0047.dat
|   |-- 0048.dat
|   |-- DATA_RADAR_INTERVALS_MEASUREID.json
|   |-- log_belt.sample-0047-0048.csv
|   |-- log_laser.sample-0047-0048.csv
|   |-- log_radar.HealthDetectionSample.0047.log
|   |-- log_radar.HealthDetectionSample.0048.log
|   |-- rr_data.csv
|-- md5.list
|-- outputs
|   |-- LABELED_SENSOR_DATA.log_belt.sample-0047-0048.csv
|   |-- LABELED_SENSOR_DATA.log_laser.sample-0047-0048.csv
|   |-- data_belt_Npeaks.0047.gauss.wavg100.json
|   |-- data_belt_Npeaks.0047.savgol.wavg100.json
|   |-- data_belt_Npeaks.0048.gauss.wavg100.json
|   |-- data_belt_Npeaks.0048.savgol.wavg100.json
|   |-- data_belt_Nscipeaks.0047.wdist100.json
|   |-- data_belt_Nscipeaks.0048.wdist100.json
|   |-- data_laser_Npeaks.0047.gauss.wavg10.json
|   |-- data_laser_Npeaks.0047.savgol.wavg10.json
|   |-- data_laser_Npeaks.0048.gauss.wavg10.json
|   |-- data_laser_Npeaks.0048.savgol.wavg10.json
|   |-- data_laser_Nscipeaks.0047.wdist10.json
|   |-- data_laser_Nscipeaks.0048.wdist10.json
```

```
|    |-- log.BULK_0000_C1.log
|    |-- log.BULK_0009.log
|    |-- plot_belt_data_and_peaks.win100__example.png
|    |-- plot_laser_data_and_peaks.win10__example.png
|    |-- results_respiration.BULK_0000_C1.json
|    |-- results_respiration.BULK_0009.json
|    |-- timestamps_radar_log_ALLDAT.log_radar.Health
           DetectionSample.0047.log
|    |-- timestamps_radar_log_ALLDAT.log_radar.Health
           DetectionSample.0048.log
|-- scripts
     |-- 0_intervals_endpoint_detection.ipynb
     |-- 1_peak_detection_belt.ipynb
     |-- 2_peak_detection_laser.ipynb
     |-- 3_run_health_detection_examples.sh
     |-- health_detection.py

4 directories, 37 files
```

There are three subdirectories: `data`, `scripts` and `outputs`. In the root directory, there is also this README file, and a file with `md5` checksums of the content files, `md5.list`.

The `data` directory contains samples of acquired data files or logs used as (primary, not derived) inputs for signal processing. The data and log samples in this directory contain a slightly redacted timestamps, nevertheless, it does not have a negative effect on showcasing the functionality of the code snippets.

The `scripts` directory contains Jupyter notebooks, Python 3 and bash scripts to process inputs from `data`.

The `outputs` directory contains outputs and logs of `scripts` executions.

A.2 Description of scripts

In the following subsections we describe the scripts purpose, inputs and outputs. With all scripts we process data samples for measurement IDs `0047` and `0048`.

A.2.1 Intervals endpoint detection

In this section we provide code snippets (script `0_intervals_endpoint_detection.ipynb`) that served to detect interval endpoints in data taking logs of the HealthDetectionSample application logs (radar data acquisition and signal processing application) discussed in Section 5.6.

Step 1: Detect full range of timestamps of data acquisition period with the radar, from the `HealthDetectionSample` application log files.

- inputs:
  - `HealthDetectionSample` application log files:

    - `/archive/data/log_radar.HealthDetectionSample.0047.log`
    - `/archive/data/log_radar.HealthDetectionSample.0048.log`

- outputs, under `/archive/outputs/`:
  - log files containing a list of timestamps from the input log files:
    - `timestamps_radar_log_ALLDAT.log_radar.HealthDetectionSample.0047.log`
    - `timestamps_radar_log_ALLDAT.log_radar.HealthDetectionSample.0048.log`

Step 2: Detect start and end timestamp for each measurement, by identifying clusters of sequential timestamps of maximal length of 70 s. The start and end timestamps for each measurement are gathered programatically, and then we manually augment the data structure with the proper measurement ID label.

- inputs, under `/archive/outputs/`:
  - log files containing a list of timestamps from the input log files:
    - `timestamps_radar_log_ALLDAT.log_radar.HealthDetectionSample.0047.log`
    - `timestamps_radar_log_ALLDAT.log_radar.HealthDetectionSample.0048.log`
- outputs:
  - A dictionary that contains the `cluster_boundaries` and `measurement_id` mapping: `/archive/data/DATA_RADAR_INTERVALS_MEASUREID.json`

Step 3: Detect measurement intervals endpoints in chest belt and distance laser measurement data: With the chest belt and distance laser measurement sample (data file or a log file), with the `cluster_boundaries` from the radar data acquisition logs, identify which chest / laser measurement corresponds with what measurement ID.

- inputs, under `/archive/data/`:
  - A dictionary that contains the `cluster_boundaries` and `measurement_id` mapping: `DATA_RADAR_INTERVALS_MEASUREID.json`
  - Chest belt data sample: `log_belt.sample-0047-0048.csv`
  - Distance laser data sample: `log_laser.sample-0047-0048.csv`
- outputs, under `/archive/outputs/`:
  - Chest belt data sample augmented with the measurement ID information (and timestamp correction discussed in Section 6.1): `LABELED_SENSOR_DATA.log_belt.sample-0047-0048.csv`
  - Distance laser data sample augmented with the measurement ID information: `LABELED_SENSOR_DATA.log_laser.sample-0047-0048.csv`

A.2.2 Peak detection in chest belt signal

In this section we discuss code snippets (script `1_peak_detection_belt.ipynb`) that served to detect peaks in chest belt data, in order to determine respiratory rate from the PLUX chest belt measurement, as discussed in Section 6.2.

In the measured data, we have information about the chest wall expansion over time, the data series shape looks like a spiky sinusoid, thus we can detect peaks in the chest wall expansion data to determine respiratory rate in this sensor signal. As discussed in Section 6.2, we use three different approaches to detect peaks:

- manual counting,
- smoothing data, moving average & local peak detection, with two different smoothing methods (Gaussian smoothing; Savitzky-Golay filter): `detect_peaks_Npeaks()`, and
- detecting peaks that are "enough apart" with `scipy`'s `find_peaks` method: `detect_peaks_Nscipeaks()`.

In this sample script, we chose `window` size 100. This script processes following inputs and provides out puts:

- inputs:
    - Chest belt data sample augmented with the measurement ID information: `/archive/outputs/LABELED_SENSOR_DATA.log_belt.sample-0047-0048.csv`
    - File that contains information about manually Counted peaks for chest belt and distance laser data: `/archive/data/rr_data.csv`
- outputs, under `/archive/outputs/`:
    - Peak detection information (number of peaks, detailed peak location data) for measurement `0047`:
        - smoothing `gauss`: `data_belt_Npeaks.0047.gauss.wavg100.json`
        - smoothing `savgol`: `data_belt_Npeaks.0047.savgol.wavg100.json`
        - `scipy` approach: `data_belt_Nscipeaks.0047.wdist100.json`
    - Peak detection information (number of peaks, detailed peak location data) for measurement `0048`:
        - smoothing `gauss`: `data_belt_Npeaks.0048.gauss.wavg100.json`
        - smoothing `savgol`: `data_belt_Npeaks.0048.savgol.wavg100.json`
        - `scipy` approach: `data_belt_Nscipeaks.0048.wdist100.json`
    - Visualization of detected peaks, comparing the 3 files for both measurement: `plot_belt_data_and_peaks.win100__example.png`

A.2.3 Peak detection in distance laser signal

In this section we provide code snippets (script `2_peak_detection_laser.ipynb`) that served to detect peaks in distance laser data, in order to determine respiratory rate from the Baumer OM70 distance laser measurement, as discussed in Section 7.2.

In the measured data, we have information about the changing chest wall distance from the laser over time, the data series shape also looks like a (even more) spiky sinusoid, thus we can detect peaks in the chest wall expansion data to determine respiratory rate in this sensor signal.

As discussed in Section 7.2, we use the same three different approaches to detect peaks:

- manual counting,
- smoothing data, moving average & local peak detection, with two different smoothing methods (Gaussian smoothing; Savitzky-Golay filter): `detect_peaks_Npeaks()` – the very same implementation as for chest belt peak detection, and
- detecting peaks that are "enough apart" with `scipy`'s `find_peaks` method: `detect_peaks_Nscipeaks()` – here we use a different parametrization of the `scipy`'s `find_peaks()` method.

In this sample script, we chose `window` size 10.

This script processes following inputs and provides outputs:

- inputs:
  - Chest belt data sample augmented with the measurement ID information: `/archive/outputs/LABELED_SENSOR_DATA.log_laser.sample-0047-0048.csv`
  - File that contains information about manually Counted peaks for chest belt and distance laser data: `/archive/data/rr_data.csv`
- outputs, under `/archive/outputs/`:
  - Peak detection information (number of peaks, detailed peak location data) for measurement `0047`:
    - smoothing `gauss`: `data_laser_Npeaks.0047.gauss.wavg10.json`
    - smoothing `savgol`: `data_laser_Npeaks.0047.savgol.wavg10.json`
    - `scipy` approach: `data_laser_Nscipeaks.0047.wdist10.json`
  - Peak detection information (number of peaks, detailed peak location data) for measurement `0048`:
    - smoothing `gauss`: `data_laser_Npeaks.0048.gauss.wavg10.json`

    - smoothing `savgol`:
      `data_laser_Npeaks.0048.savgol.wavg10.json`
    - `scipy` approach:
      `data_laser_Nscipeaks.0047.wdist10.json`
  - Visualization of detected peaks, comparing the 3 files for both measurement: `plot_laser_data_and_peaks.win10__example.png`

A.2.4 The `health_detection` application code

In this section we present code of the `health_detection` application to study parametric model for respiratory rate (RR) detection, discussed in Section 5.4.

We provide a wrapper script `3_run_health_detection_examples.sh` to run the RR detection for measurement IDs `0047` and `0048`, for parametric model configurations discussed in Section 5.5:

- the most suitable parametrization for RR detection with CFAR peak detection model (5.19), with configuration `BULK_0000_C1`, and
- the most suitable parametrization for RR detection with CWT peak detection model (5.20), with configuration `BULK_0009`.

For this example, we processed following files:

- inputs: radar data files:
  - `0047`: `/archive/data/0047.dat`
  - `0048`: `/archive/data/0048.dat`
- outputs, under `/archive/outputs/`:
  - respiration result files:
    - `BULK_0000_C1`: `results_respiration.BULK_0000_C1.json`
    - `BULK_0009`: `results_respiration.BULK_0009.json`
  - execution log files:
    - `BULK_0000_C1`: `log.BULK_0000_C1.log`
    - `BULK_0009`: `log.BULK_0009.log`

The parametrization used is listed in `3_run_health_detection_examples`. The `health_detection` script allows command line parameters listed in Table A.1.

A.3 The `HealthDetectionSample` application `Dockerfile`

We containerized the `HealthDetectionSample` application as discussed in Section 5.1. The `Dockerfile` is presented below.

CLI option	**Default value**	**Purpose**
`-L, --label`	BULK-TEST	Label to distinguish bulk data processing run
`-D, --record_dir`	./data/	Directory to store radar data
`-F, --filename`	0047.dat	Filename with the radar data
`-P, --peak_detection_method`	cfar	Peak detection method (cfar/cwt). Choices: cfar, cwt
`-T, --cfar_num_train`	50	Number of training cells for CFAR
`-G, --cfar_num_guard`	200	Number of guard cells for CFAR
`-R, --cfar_false_alarm_rate`	0.001	False alarm rate for CFAR
`-U, --cwt_guard`	150	Number of guard cells for CWT
`-s, --cwt_scale_max`	80	Max of range for CWT scales
`-W, --cwt_wavelet`	gaus1)	CWT wavelet. Choices: `pywt.wavelist(kind='continuous')`
`-r, --gradient_peak_threshold_resp`	0.01	Gradient peak threshold for respiration rate detection
`-g, --gradient_peak_threshold_heart`	0.1	Gradient peak threshold for heart rate detection
`-V, --max_variance`	100.0	Maximum allowed variance for target range bin changes
`-O, --outlier_remove_threshold`	1.5	Threshold to remove occasional false detection points
`-o, --outlier_variance_remove_threshold`	1.5	Variance threshold for outlier removal
`-C, --remove_outliers_center`	mean	Center for outlier removal. Choices: mean, median.
`-c, --remove_variance_outliers_center`	median	Center for variance outlier removal. Choices: mean, median.
`-k, --kalman_init_sample_size`	10	Kalman filter initialization sample size
`-S, --save_data_sample`	False	Save a dump of data as a sample to visualize later

Table A.1: The `health_detection` application command line parameters.

```
FROM ubuntu:22.04
RUN apt-get update && \
  apt-get install -y sudo  cmake  build-essential  git
     vim  libboost-all-dev  nlohmann-json3-dev
     libeigen3-dev    && \
  rm -rf /var/lib/apt/lists/*
# Define mountable directories.
VOLUME ["${CODE_DIR}", "${DATA_DIR}"]
# Add user
RUN useradd -rm -d /home/${WRK_USERNAME} -s /bin/bash -g
   root -G sudo -u 1001 ${WRK_USERNAME}
# sudo needed for interactive debugging & setup/chown
RUN echo '${WRK_USERNAME} ALL=(ALL) NOPASSWD: ALL' >>
   /etc/sudoers
# Setup directories & their ownership
RUN chown -R ${WRK_USERNAME} /home/${WRK_USERNAME}
RUN chmod 755 /home/${WRK_USERNAME}
RUN mkdir -p ${CODE_DIR} ${DATA_DIR}
RUN chown -R ${WRK_USERNAME} ${CODE_DIR}
RUN chmod 755 ${CODE_DIR}
RUN chown -R ${WRK_USERNAME} ${DATA_DIR}
RUN chmod 755 ${DATA_DIR}
RUN chown -R ${WRK_USERNAME} /home/${WRK_USERNAME}/wrk
RUN chmod 755 /home/${WRK_USERNAME}/wrk
# Switch over to the user environment
USER ${WRK_USERNAME}
# Define the working directory.
WORKDIR /home/${WRK_USERNAME}
# Define default command.
CMD ["/bin/bash"]
```

A.4 Distance laser data acquisition

In this section we provide code snippet that served to acquire data from the Baumer OM70 distance laser, as discussed in Section 7.1. We used the `pySerial` library [23] to configure the distance laser and read out data from it.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import datetime
import serial


### Bauder OM70 laser commands
CMD_LOCK_DEACTIVATED = b":01W010;0;E9C3\r\n";
CMD_DEVICE_INFO = b":01R002;3955\r\n";
CMD_GET_BAUDRATE = b":01R006;F957\r\n";
CMD_SET_BAUDRATE = b":01W006;1;C1FF\r\n";
CMD_GET_MEASUREMENT = b":01R021;09F4\r\n";
```

```
CMD_GET_ALL_MEASUREMENTS = b":01R022;F9F4\r\n";


def cmd(dev, command, caption):
    """
    Executes a command on a device and prints the command
       with its caption before execution and the output
       after execution.

    Parameters:
    - dev: The device connection object on which the
       command will be executed. This object should have a
       write() method to send the command and a
       readlines() method to read the output.
    - command: A string representing the command to be
       executed on the device.
    - caption: A brief description or identifier for the
       command, used for logging and output purposes.

    Returns:
    - A list of strings representing the lines of output
       from the executed command.

    The function first logs the command to be executed
       along with its caption, then sends the command to
       the device using the device's write() method. It
       waits for the command to execute and reads the
       output using the device's readlines() method.
       Finally, it logs and returns the output.
    """
    dev.write(command)
    res = dev.readlines()
    return res


if __name__ == "__main__":
    ### Initiate the device
    dev = serial.Serial('/dev/ttyUSB0', 57600,
       timeout=0.1, parity=serial.PARITY_EVEN)

    ### Set the device's baudrate
    cmd(dev, CMD_SET_BAUDRATE, "CMD_SET_BAUDRATE")

    ### Read the device's baudrate
    cmd(dev, CMD_GET_BAUDRATE, "CMD_GET_BAUDRATE")

    ### Deactivate the lock
    cmd(dev, CMD_LOCK_DEACTIVATED, "CMD_LOCK_DEACTIVATED")

    ### Get the device information
    cmd(dev, CMD_DEVICE_INFO, "CMD_DEVICE_INFO")
```

```
### Read out a measurement
cmd(dev, CMD_GET_MEASUREMENT, "CMD_GET_MEASUREMENT")

### Measurement loop
while True:
    ### Data point timestamp
    ts = datetime.datetime.utcnow()
    ### Data point measurement string
    res = cmd(dev, CMD_GET_MEASUREMENT,
       "CMD_GET_MEASUREMENT")
    ### Log the data point value
    print(f"{ts}: {res}")
```