Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

My Tam Lam

Automatic Detection of Air Emboli in The Cerebral Circulation in Newborns By Ultrasound Doppler Technique

Master's thesis in Applied Physics and Mathematics
Supervisor: Hans Torp
Co-supervisor: Jon Andreas Støvneng
February 2022

NTNU
Norwegian University of
Science and Technology

My Tam Lam

Automatic Detection of Air Emboli in The Cerebral Circulation in Newborns By Ultrasound Doppler Technique

Master's thesis in Applied Physics and Mathematics
Supervisor: Hans Torp
Co-supervisor: Jon Andreas Støvneng
February 2022

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU
Norwegian University of
Science and Technology

MY TAM LAM

AUTOMATIC DETECTION OF AIR EMBOLI IN THE CEREBRAL CIRCULATION IN NEWBORNS BY ULTRASOUND DOPPLER TECHNIQUE

AUTOMATIC DETECTION OF AIR EMBOLI IN THE CEREBRAL CIRCULATION IN NEWBORNS BY ULTRASOUND DOPPLER TECHNIQUE

MY TAM LAM

NTNU

Master Thesis in Applied Physics and Mathematics
Supervisor: Hans Torp / Jon Andreas Støvneng

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Fall 2021

ABSTRACT

Infants who are born with heart defects may need catheter intervention or heart surgery as treatment. Such interventions can cause air bubbles to be introduced into the bloodstream. These events can cause clogged arteries or other more severe issues for the patient and can potentially be fatal. Consequently, a method for detecting the extent and locations of air bubbles situated in the bloodstream is crucial for the treatment of the complication that may result from air embolism. By utilizing the NeoDoppler system for the ultrasound analysis of the cerebral blood flow, accompanied with the EarlyBird software, higher intensities of the ultrasound signal can be recognized as air bubbles by analyzing the Color M-Mode image and Doppler spectrogram.

In clinical research, air bubbles are detected by manually searching the Color M-Mode image and Doppler spectrogram for areas of higher intensities. This process is very time-consuming and can result in mixed results depending on which criteria are used for the detection and by whom the detection is performed. Preferably, the detection process should also be performed multiple times and by different people to verify the result. In this project, the aim is to develop an algorithm that performs automatic detection of air emboli in the cerebral circulation to increase the efficiency of the detection process and assist the human observer in the clinical environment.

The algorithm starts the detection process by prepping the ultrasound signal with a lowpass filter. Furthermore, the algorithm is checked for artifacts. Then, the actual detection process is performed by analyzing one depth at a time. For each depth, the background signal is estimated. Each point of the recording with an intensity higher than a chosen threshold above the background signal will be detected. Then, the detections will go through a correction process, as some of the bubbles detected might be noticeable over multiple depths.

For the training and testing of the algorithm, there were in total 650 recordings included in this project. For the training of the algorithm, 68 of the recordings were used, involving one patient during transcatheter intervention and one patient during heart surgery. Manual detections of the recordings are also included for the ability to measure the performance of the algorithm. The training set had 75 detections, where 23 of the detections were false positive, and 78 detections were missed in comparison with manual detections.

For the test set, a total of 582 recordings was included, consisting of 26 different patients, 14 patients during catheter intervention, and 12 patients during heart surgery. In total, 351 detections were obtained,

where 241 of these were false detections and 259 were missed detections.

The strict criteria for the detection of air bubbles used in this algorithm might be the reason for the large amount of missed detections, as well as artifacts present in the ultrasound signal and an inagreement of the emboli-to-blood ratio of the detections made by the algorithm and the emboli-to-blood ratio estimated for the manual detections included for comparison.

SAMMENDRAG

Når spedbarn er født med hjertesvikt eller en annen form for defekt i hjertet kan det være nødvendig med kateterintervensjon eller hjertekirurgi. Slike inngrep kan føre til at luftbobler blir introdusert inn i blodstrømmen. En slik situasjon kan føre til tette arterier eller andre mer alvorlige komplikasjoner, og kan potensielt være dødelig. Av den grunn er det et stort behov for en metode for deteksjon av omganget av og posisjonene til luftbobler som befinner seg i blodårene, ettersom dette kan være til stor hjelp til behandlingen av komplikasjonene som kan medfølge luftembolisme. Ved analyse av analyse av Color M-mode-bildet og Doppler spektrogrammet visualusert gjennom systemet NeoDoppler og programvaren EarlyBird, kan ekkosignaler av bobler gjenkjennes som høyere intensiteter i blant ultralydsignaler av blodet.

I klinisk forskning detekteres luftbobler ved å manuelt søke etter høyere intensiteter i Color M-mode-bildet og Doppler spektrogrammet. Dette er en prosess som ikke bare er svært tidkrevende, men som også kan resultere i varierende resultater, ettersom deteksjonen kan baseres på ulike kriterier og variere avhengig av hvem som utfører deteksjonen. Helst skal også deteksjonen utføres opp til flere ganger og av ulike personer for en verifisering av resultatet.

Dette prosjektet har som mål å utvikle en algoritme for automatisk deteksjonen av luftbobler. Målet er å lage en algoritme som gjør prosessen av deteksjon mer effektiv, slik at den muligens kan assistere the menneskelige observatøren i the kliniske settinger.

Algoritmen starter med å lavpassfiltrere ultralydsignalet for raske fluktuasjoner i signalet. Videre sjekker algoritmen ultralydsignalet for artefakter. Deretter kan selve deteksjon begynne, og gjøres ved analyse av en dybde om gangen. For hver dybde vil bakgrunnssignalet bli estimert, slik at en deteksjonsterskel kan bli satt. Ethvert tidspunkt av lydsignalet med en amplitude over deteksjonsterskelen vil bli detektert. Videre vil algoritmen gjennomføre en korrigering av deteksjonene, ettersom boblenesignalene kan strekke seg over flere dybder og føre til flere deteksjoner av samme boble.

For treningen og testingen av algoritmen er det inkludert 650 ultralydopptak. For treningssettet er 68 av lydopptakene inkludert, og involverer to pasienter, én fra kateterintervensjon og én fra hjertekirurgi. Manuelle deteksjoner er også inkludert i prosjekter, ettersom de skal brukes for en sammenlikning med de deteksjonene algoritmen utfører og bidra til å avgjøre deteksjonsevnen til algoritmen. Treningssettet hadde totalt 75 deteksjoner, der 23 av deteksjonene var falske positive, og 78 var savnede deteksjoner som man kunne finne blant de manuelle deteksjonene.

For testsettet utførte algoritmen var 582 ultralydopptak inkludert, der opptakene var fra 26 forskjellige pasienter. 14 av pasientene var fra kateterintervensjon og 12 pasienter var fra hjertekirurgi. Det ble totalt detektert 351 deteksjoner i testsettet. Av disse var 241 deteksjoner falske, og 259 av de manuelle deteksjonene var savnet blant algoritmenes deteksjoner.

Hovedårsakene til den store andelen med falske deteksjoner og savnede deteksjoner inkluderer strenge kriterier for hva som kan gjenkjennes som et boblesignal, samt artifakter som er tilstede i opptaket. videre kan også uoverenstemmelsen mellom kalkulert emboli-til-blod ratio (EBR) av de automatiske deteksjonene og de manuelle deteksjonene forklare et svakere resultat.

PREFACE

This report presents the final results of the Master's thesis related to the course TFY4900 - Physics, Master's Thesis. The report is the final work in the Master's degree program, Applied Physics and Mathematics, at the Norwegian University of Science and Technology, NTNU. It is a continuation of the work done in the course TFY4510, where we decide on a specialization topic and investigate theory and methods related to this topic. For the specialization topic, the theory of the ultrasound scattering of blood and bubbles was described and implemented to obtain an expression describing the relation between the signal from the emboli to the background signal and a method for estimating the emboli-to-blood ratio through simulations. In this thesis, the work continues by implementing the theory to construct an automatic detection algorithm of air emboli in newborns. The thesis was written during the fall semester of 2021 in collaboration with the Department of Circulation and Medical Imaging.

A professor at NTNU has guided the project closely. I want to give a special thanks to my supervisor, professor Hans Torp at the Department of Circulation and Medical Imaging, for the thorough and constructive guidance these two last semesters, and for always being of such availability, even during the Covid-19 lockdown. I would also like to thank Martin Leth-Olson and Siri Ann Nyrnes at St. Olavs Hospital for their contribution to data collection and for their inputs and thoughts on the project.

Lastly, I wish to thank my family and friends for all their love and support.

Oslo, February 2022
My Tam Lam

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ABBREVIATIONS

CHD	Congential Heart Disease
CI	Catheter Intervention
CMD	Color M-Mode Doppler
CPB	Cardiopulmonary Bypass
CW	Continuous Wave Doppler
dB	Decibel
EBR	Emboli-to-Blood Ratio
FFT	Fast Fourier Transform
HBR	Hits-to-Blood Ratio
HCT	Hematocrit
HITS	High Intensity Transient Signals
HS	Heart Surgery
IQ	In-Phase Quadrature Signal
MEBR	Measured Emboli-to-Blood Ratio
PW	Pulsed Wave Doppler
RBC	Red Blood Cell
RMS	Root Mean Square Value
SNR	Signal-To-Noise Ratio
TCD	Transcranial Doppler
US	Ultrasound

SYMBOLS

a	Radius
c	Speed of sound
d	Diameter
D	Distance
F	Force
f	Frequency
I	Acoustic Intensity
K	Bulk Modulus
k	Acoustic Wavenumber
m	Mass
N_p	Number of Samples
p	Pressure
P	Acoustic Power
R	Resistance
r	Radius
s	Spring Constant
S	Surface Area
T	Stress Tensor
u	Particle Velocity
V	Volume
Z	Acoustic Impedance
γ	Specific Heat
δ	Damping Constant
ϵ	Backscattering Coefficient
η	Shear viscousity
θ	Isonation Angle
κ	Polytropic Index
λ	Wavelength
ρ	Mass Density
σ	Scattering Cross Section
ξ	Radial Displacement
ψ	Velocity Potential
Ω	Normalized Angular Frequency
ω	Angular Frequency

Part I

INTRODUCTION

1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Infants with heart defects, such as congenital heart disease (CHD), undergoing surgery or intervention treatment are at risk of getting air bubbles in their bloodstream. These events occur rarely but can cause severe damage and potentially be fatal events. Being able to detect these occurrences is therefore crucial. Developing a method for the automatic detection of air emboli can help save time and create an overview of the air bubbles present in a patient's blood vessels and help the clinicians decide what procedures to proceed with.

Of infants with CHD, around 25% have critical heart defects. This often results in the infants needing intervention or surgery within the first year of life. When infants with CHD go through transcatheter cardiac intervention or surgery, there is a risk of the formation of gaseous emboli in the blood vessels. [13] The emboli can potentially clog the arteries or other more severe complications. If the emboli are in the cerebral circulation, they can cause severe neurological deficit. In the worst case, they can lead to death.

Because it is low cost, easy to use, and non-invasive, diagnostic ultrasound is commonly used in medical diagnosis. An image of the body's interior can be composed based on the scatter and reflection of the ultrasound waves. With ultrasound Doppler, the movement of body fluids can be measured, enabling blood flow analysis with any air bubbles that may be situated in the blood vessel.

In experimental research, Transcranial Doppler (TCD) has been used for the detection of emboli for decades. [13] The detection of bubbles has traditionally been carried out by clinicians by manually searching for higher intensities in the color m-mode (CMD) and Doppler spectrum. Not only can the results vary depending on who is performing the detection and on what criteria has been used for distinguishing the air bubbles from the blood signal, but the process of manually detecting can also be time-consuming. Preferably, the process should also be verified by counting multiple times and by different clinicians. Thus, this is not a process that is possible to do in real-time, creating the need for a method to detect air emboli automatically.

With an algorithm for automatic detection of air emboli, time can be saved, and an overview of the emboli present can be obtained. Furthermore, it will also reduce the possible source of error depending on who is performing the detection. An algorithm will objectively detect bubbles in the same manner, as the criteria for detection can

be set equal for all procedures. The advantages of an algorithm for automatic emboli detection include a more effortless and less time-consuming detection process in clinical research. With an algorithm of fast running time, the automatic detection can potentially enter routine clinical practice and guide additional diagnostic and therapeutic decisions.

1.1.1 *Literature Review*

The attempt to improve the methods for detection of air emboli through the use of ultrasound is a process that has been going on for several decades. There are a lot of different methods that have been tested out, as the motivation for achieving automatic detection of emboli is of a great deal.

The use of TCD in the research on embolism in the cerebral circulation dates at least back to 1993, when Markus, Loh, and Brown performed their study on "Computerized Detection of Cerebral Emboli and Discrimination from Artifact Using Doppler ultrasound." They used TCD for the detection of circulating cerebral emboli in both sheep and patients. Their detection algorithm was based on the studied characteristics of a bell-shaped increase in the relative power amplitude associated with emboli and the differing characteristics of the symmetrical bidirectional intensity increase associated with artifacts. They found that the detection of emboli and artifacts was successful. However, distinguishing artifacts from emboli was more challenging and gave a less promising result. [14]

In 2000, Cullinane et al. performed the study "Evaluation of New Online Automated Embolic Signal Detection Algorithm, Including Comparison With Panel of International Experts." In the study, they have applied TCD from the middle cerebral artery for an evaluation of an automatic detection algorithm. The algorithm uses a conventional fast Fourier transform (FFT) with a Hanning function and is a single-gated method, meaning that the algorithm only analyzes the signal at one depth. The background signal is calculated as an average over the FFT with 750 seconds before and after each event. A 2D median filter finds the background level, and all events are compared to this level. Any event with an intensity of 3 dB or higher than the background level is further analyzed as embolic signal candidates. The further analysis is based on different characteristics in the frequency domain. From Cullinane et al.'s previous work, they have found that time or frequency disorder tends to be low for emboli and high for artifacts. These results are implemented into their study to distinguish embolic signals from artifacts. [6]

Guepie et al. tests out a different method for the automatic detection of emboli in their study on "Sequential emboli detection from ultrasound outpatient data" from 2019. Their method involves using

a single-gated detection algorithm followed by a classification algorithm. The detection algorithm is based on the ratio between the high intensity transient signals (HITS) and the estimated blood flow power, as they call HITS-to-blood ratio (HBR). Any HBR with a value above a set amount of decibels (dB) will be detected. Furthermore, eight features are calculated using the HITS signals and the Doppler spectrogram. These features are further given to the classification algorithm, consisting of three different supervised machine learning algorithms.[8]

Different from the two previously mentioned studies, Kjelsaas's study on *Detection of Air Emboli in the Brain of Neonates by Ultrasound Doppler* from 2020 is based on a multi-gated detection algorithm, meaning that the algorithm involves several depths for the analysis. The algorithm detects bubbles based on analysis of the power signal of the ultrasound recording, given depth by depth. Kjelsaas estimates the background signal of the given depth by finding the median of the power signal. Then, a threshold is set by the sum of the background signal and the chosen emboli-to-blood ratio (EBR) in decibels (dB). Kjelsaas tested out EBR values from 8 dB to 9.5 dB. Any part of the power signal with higher amplitude than the threshold will be detected. Then, the number of detections will be corrected for duplicates, as some bubbles may appear in several depths. [11]

The recordings included in Kjelsaas's study are gathered from Leth-Olsen et al.'s study on *Detection of Cerebral High Intensity Transient Signals by NeoDoppler During Cardiac Catheterization and Cardiac Surgery in Infants* where Leth-Olsen et al. has manually detected bubbles in the ultrasound recordings, by looking at intensity increases of the CMD and by analysis of the Doppler spectrogram. The recordings are of 31 patients scheduled for transcatheter intervention or heart surgery. [13] Of these, Kjelsaas included 405 recordings, with 16 patients during catheter intervention and two patients during heart surgery. The test set of the algorithm resulted in 82.4% of the detections being false positives. The

In order to evaluate the results of the algorithm, Kjelsaas has also included detections performed manually by Leth-Olsen et al. in her project. From the comparison of the test set with the manual detections, Kjelsaas found that 82.4% of the algorithm's detections were false positives. These were primarily due to cyclic variations with the heart frequency, which were not present in the training set for the algorithm, possibly being the cause of the algorithm's inability to compensate for these effects. Possible future improvements included implementing functions that handle the different situations causing false positives or limiting the algorithm to only aim to detect larger bubbles.

1.2 AIMS OF STUDY

This study aims to develop an algorithm for detecting air emboli in the cerebral circulation. An automatic method for the detection of air embolism can help clinicians save time when searching for air bubbles and possibly, if promising enough, replace the human observer in the clinical environment. Hence, a more specific goal is to develop an algorithm of a short enough running time in order to be able to use the algorithm in real-time.

For this project, recordings and associated data on manual detections will be included from Leth-Olsen et al.'s study on *Detection of Cerebral High Intensity Transient Signals by NeoDoppler During Cardiac Catheterization and Cardiac Surgery in Infants*, similar to Kjelsaas. Thus, further aims of this study are to improve the results from *Detection of Air Emboli in the Brain of Neonates by Ultrasound Doppler* by Kjelsaas, by creating an algorithm that can handle the effects that caused Kjelsaas errors in her result. That includes a focus on detecting bubbles of larger sizes and a solution for the algorithm to handle situations with cyclic variation with the heart frequency.

1.3 OUTLINE OF THE REPORT

This report consists of six chapters. Beginning with the introduction, the background and motivation for this project are presented, along with the aims of the study. Then, the theory is presented to form the basis of the project. The third chapter presents and describes the algorithm developed in detail. Furthermore, the equipment, recordings, and methods used for developing and evaluating the algorithm are described. The results of both the training and the testing of the algorithm are presented in the next chapter, followed by a discussion of the results. Finally, conclusions are drawn in the last chapter, along with possible future improvements of the algorithm.

Part II

THEORY

2 THEORY

This chapter will present the theory pertinent to the project. Starting off the chapter is the theory on the possible complications of air embolism. Then, the ultrasound technology used is presented before discussing the acoustic effects of blood and bubbles.

2.1 AIR EMBOLI COMPLICATIONS

When air enters the cerebral circulation, it can potentially obstruct the blood flow, causing severe damage and, in the worst case, be fatal. A plug of clot, fat, air or other material not consisting of blood that obstructs the blood flow in the blood vessel is defined as an embolism. [17] Air embolism is one of the different types of embolism and consists of one or more bubbles of gas enclosed within the blood vessels and occurs when air is introduced to the vascular system.

Air embolism can occur due to various circumstances. These range from diving and childbirth to trauma and surgeries. In general, air emboli can only occur when a connection exists between air and the vascular system. For instance, this happens when the veins or arteries are exposed to air. For the air to flow into the vascular system, there must be a pressure gradient enabling the flow of air into the blood vessels. The pressure gradient force naturally drives air from higher pressure areas to lower pressure areas. [7] Generally, the pressure in blood vessels is higher than the atmospheric pressure. However, in some areas of the body, the pressure is lower than in the atmosphere, typically the head and neck region. Consequently, these areas will be more prone to air emboli during intervention [7]. There is a range of clinical situations where air can enter the blood flow, including trauma, barotrauma, central line placement, interventional radiology procedures, central line placement and removal, and some types of surgery such as cardiac and neurosurgery [16].

There are three different types of air embolism. When air bubbles enter the veins, they are called venous emboli. These are generally of no harm, as they are usually stopped at the lungs. However, venous emboli always have the potential to become arterial emboli if there exists a connection between the two systems, for example, when there is a hole in the septum of the heart, and the pressure gradient allows for a flow of air between the two systems. The arterial emboli is a more harmful type of air emboli and are emboli situated in the arteries. As the arteries carry blood away from the heart and to the rest of the body, emboli in these vessels might prevent oxygenated blood from

reaching target organs. This can cause ischemia, which is a deficiency in tissue that can cause reduced oxygen supply to the tissue or infarction in any organ with limited blood supply. [23] Furthermore if the arterial emboli reach the cerebral circulation, these emboli are rather referred to as cerebral emboli, as they are a far more dangerous type of air emboli. A blockage of cerebral arteries obstruct the essential blood supply to the brain, which is crucial for its functionality and vital processes. [16] [2]

The complications due to air in the vascular system are determined by the air emboli's volume, rate, and location. The physiological effects may range from asymptomatic to cardiovascular disease and death. An air embolus might obstruct the blood flow. Oxygen is transported around the body through the blood vessels. If blood flow is blocked due to air emboli, oxygen transportation is prevented. The body's tissues and organs need oxygen. Consequently, the prevention of oxygen supply might cause the death of body tissue. In addition, prevention can cause the organ to lose some or all of its function. If the prevention is affecting the vital organs of the body, such as the brain and the heart, the situation can become extremely harmful. Air embolism in the cerebral circulation can cause a neurological deficit. The neurological symptoms included focal motor deficits, sensorium changes, and visual and sensory deficits. Worst-case scenarios include coma, stroke, and death. [20]

2.2 ULTRASOUND TECHNOLOGY

Sound waves with frequencies above 20 kHz are known as ultrasonic sound waves, or more commonly known as ultrasound (US). Ultrasound can be applied in many different fields, and in this project, the application of ultrasound in medicine will be relevant. Ultrasound is commonly used in medical diagnosis because of its low cost, applicability, and non-invasiveness. Diagnostic ultrasounds typically use sound waves with a frequency between 2 to 10 MHz. Through the scatter and reflection of sound, an image of the body's interior is formed.

The ultrasound waves are produced in part called the transducer. The transducer is a part of the ultrasound device and contains piezoelectrics, an active element made of special ceramic crystal materials. As voltage is applied, the thin metal electrodes surrounding the piezoelectrics will cause a mechanical displacement in the crystal, producing sound waves. As the sound waves propagate through layers of different acoustic impedance, the sound waves will be partly reflected and transmitted. When the returning echoes hit the transducer, the piezoelectric material will cause the echoes to turn back into electric signals. Following, the electric signals will enter a chain of signal pro-

cessing, transforming the electric signals into an ultrasound image. [1]

2.2.1 *Ultrasound Doppler*

Ultrasound Doppler is a type of ultrasound technique, where US waves are used to image objects of movement—measuring sound waves that are reflected from moving objects. This technique is based on the Doppler effect.

The Doppler effect was discovered by Christian Andreas Doppler, explaining the change of the perceived frequency of an emitted wave due to a moving source or moving observer. [22] Given that the propagation velocity is constant, the effect is valid for both electromagnetic waves and sound waves. As the source moves in the direction of the wave propagation, an increase of the frequency in the direction of wave propagation will occur. Similarly, there will be a decrease in the frequency in the opposite direction. The Doppler effect has become of practical importance in many fields, including ultrasound. By employing the Doppler effect in the field of ultrasound, it is possible to create ultrasound images of the movement of tissue and body fluids, such as blood flow.

There are different types of Ultrasound Doppler: Continuous Flow Doppler CW, Pulsed Wave Doppler PW, Color Flow Doppler, and Power Doppler. The most common type of Ultrasound Doppler for quantitatively imaging is CW and PW, while Color Flow Doppler is more suited for qualitatively using. CW Doppler emits a continuous ultrasound wave during the ultrasound. Consequently, the transducer of a CW Doppler only uses half of the transducer for transmission, allowing the other half to receive returning echoes continuously. As the transducer is continuously transmitting and receiving sound waves from all depths at all times, the CW Doppler has no range of resolution. Oppositely, the PW Doppler uses the whole transducer surface for transmission of sound waves and for receiving returning echoes, as it emits short pulses of sound waves. In addition, this opens up the possibility to image at specific depths. [18]

With ultrasound Doppler being able to image moving objects, it has become an essential tool in various clinical applications. In this report, the PW Doppler has been applied for the observation of the cerebral circulation and any air bubbles that might be situated within it. Doppler signal from blood and bubbles of the cerebral circulation is obtained by scanning through the cranium, also called transcranial Doppler ultrasound (TCD).

2.2.2 *Doppler Signal from Bubbles and Blood*

With the application of PW Doppler, the continuous measurement of blood flow is possible. By placing the transducer probe above an artery at an angle different from 90 degrees, blood and any air bubbles can then be observed. With the angle ϕ of the path the bubble is passing the ultrasound beam at and the diameter dt of the ultrasound beam being known, as shown in Figure 2.1, the velocity of the bubble can be calculated. For each pulse the ultrasound beam sends while the bubble travels inside the ultrasound beam, the sound wave will be reflected back as an echo to the transducer, as shown in Figure 2.1.

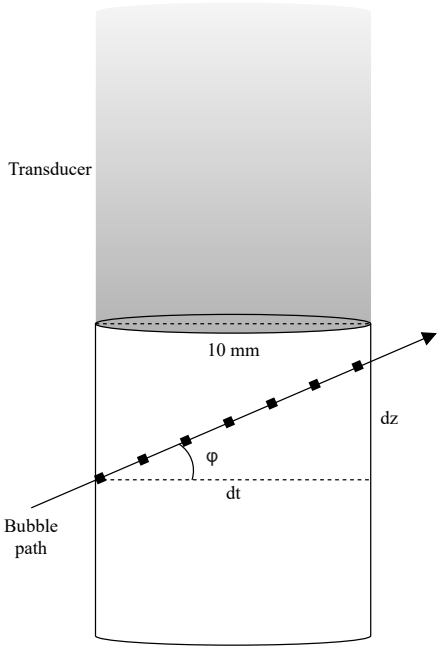


Figure 2.1: The path of a bubble crossing the ultrasound beam from the transducer.

By observing the IQ signal, one can observe the magnitude of the intensity of the Doppler signal of the bubble being more prominent than for the Doppler signal of the surrounding blood. This is displayed in Figure 2.2, where the real part of an IQ signal is displayed. The cause of the difference in acoustic intensities is due to the blood and air having a different acoustic impedance. A medium's acoustic impedance (Z) is defined as the product of the density (ρ) of the medium and the speed of sound (c) in the medium.

$$Z = \rho \cdot c \tag{2.2.1}$$

When encountering an interface of different acoustic impedance, sound waves are scattered. The scattering will cause the sound waves to be partly reflected and partly transmitted, according to how big the dif-

ference in acoustic impedance is. The greater the difference in acoustic impedance is, the bigger the reflection fraction.

$$\text{Reflection fraction} = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1}\right)^2 \tag{2.2.2}$$

With the acoustic impedance of blood and other bodily tissues being much more similar than the acoustic impedance of air, a larger portion of the sound waves will be reflected by air than by blood. [19] Thus, the signals from bubbles can be differentiated from the signals from blood because of higher intensities in the signal.

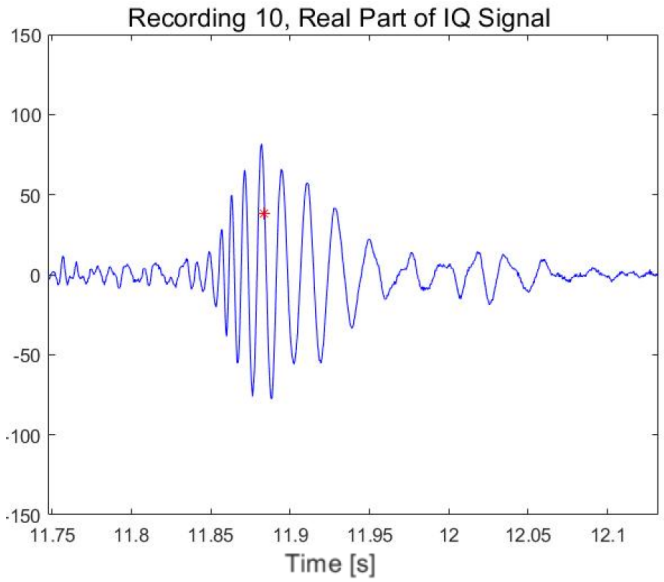


Figure 2.2: Doppler signal of blood containing an air bubble, where the red mark indicates the maximum intensity of the signal. Reprinted from Kjelsaas on her work on *Detection of Air Emboli in the Brain of Neonates by Ultrasound Doppler*. [11]

2.3 ACOUSTICS

Even with known details about the structure of the biological media, the scattered signal from an incident ultrasound signal can be difficult to fully interpret, as the structure of the biological media is very complex. The theory presented in the following section is gathered from previous work on *Estimation of Air Bubbles in The Cerebral Circulation in Newborns By Ultrasound Doppler Technique*. [12]

For an incident plane sound wave, the time-averaged total power, $\bar{P}_t$, is the sum of the scattered and absorbed power, and defines the total scattering cross-section. The total scattering cross-section, σ_t is defined as the time-averaged total power, $\bar{P}_t$, per time-averaged incident intensity, $\bar{I}$, and is equivalent to the area of the incident plane

wave that contains an amount of incident power equal to the time-averaged total power.

$$\sigma_t = \frac{\bar{P}_t}{\bar{I}} \tag{2.3.1}$$

In the same way, the scattering cross section is defined as the time-averaged scattered total power, $\bar{P}_s$, per time-averaged incident intensity, $\bar{I}$.

$$\sigma(\theta, \phi) = \frac{\bar{P}_s(\theta, \phi)}{\bar{I}} \tag{2.3.2}$$

The differential scattering cross section gives the variation of scattered power per unit solid angle. Thus, the differential scattering cross section of 180 degrees defines the backscattered cross section. [5]

$$\sigma_b = \sigma(\pi, 0) = \frac{\bar{P}_s(\pi, 0)}{\bar{I}} \tag{2.3.3}$$

A method for modelling spherical and cylindrical scatters by acoustic waves, is the boundary method, initially used by Rayleigh. The simple model for the scatter of small objects is called Rayleigh scattering, and is valid for objects with dimensions much smaller than the acoustic wavelength

$$d \ll \lambda \quad \text{or} \quad ka \ll 1, \tag{2.3.4}$$

where d is the particle diameter, λ is the wavelength of the incident wave, $k = \frac{2\pi}{\lambda}$ is the acoustic wavenumber, and a is the particle radius. [9] Oppositely, particles with a diameter close to the wavelength will experience a scattering influenced by resonance and other scattering effects. This scattering-type is referred to as anisotropic or non-Rayleigh scattering. [15] From the book *The Theory of Sound*, Rayleigh gives a model for the scatter of sound from small objects. [21] Formulated as the scattering cross-section, the model has the expression

$$\sigma = 4\pi a^2 (ka)^4 \left[\left(\frac{K - K_0}{3K} \right)^2 + \frac{1}{3} \left(\frac{\rho - \rho_0}{2\rho + \rho_0} \right)^2 \right] \tag{2.3.5}$$

where ρ, ρ_0 are the densities of the particle and the surrounding medium, K, K_0 are the bulk moduli of the particle and the surrounding medium. The bulk modulus gives the resistance the substance have to compression. From Equation (2.3.5), the scattering cross section's dependence on the particle size and frequency is shown. The scattering cross-section is proportional with a^6 and k^4, which means that the cross-section increases by the frequency to the 4^{th} power. [9]

With the Rayleigh approximation of scattering being valid for objects of small sizes compared to the acoustic wavelength, the approximation is suited for the scattering of blood cells and air bubbles in

the blood vessels. However, as the approximation excludes resonance and sound absorption, it might not be the most suited model for describing the scattering of bubbles. Yet, the Rayleigh model for scattering can still give an idea of the scattering mechanisms and properties causing scatter of sound and exhibits how the bubble is a powerful scatterer.

2.3.1 *Scattering of Blood*

At lower frequencies, and consequently larger wavelengths, the dimensions of a red blood cell RBC will be much smaller than the wavelength. Under these circumstances, Rayleigh scattering is a good approximation for the RBC. The scattering cross section is then given by (2.3.5). With the RBC having a compressibility of $K = 0.3911\ \text{GPa}^{-1}$ and a density of $\rho = 1078\ \text{kg/m}^3$, and the surrounding fluid being plasma and having a compressibility of $K_0 = 0.4421\ \text{GPa}^{-1}$ and a density of $\rho_0 = 1004.6\ \text{kg/m}^3$, the scattering cross section is given as in (2.3.6), where a is the radius of the RBC.

$$\sigma = 4\pi a^2 (ka)^2 \cdot 2.06942 \cdot 10^{-3} \tag{2.3.6}$$

A measure for the number and size of RBC in the blood is given by hematocrit HCT levels, and give the percentage of the RBC in the total blood volume of a person. For men, a normal HCT level is around 40 - 58 %, and for women, it is around 36 - 48%. [3]

When estimating the scattering by the distribution of RBC in blood volumes, there are multiple factors affecting the scattering. These include the whether or not the distribution is random or organized, because an organized distribution can cause interference. Another factor is the dimensions of the scatterers, if there are scatterers of different size, shapes, or acoustic properties. In addition, if the scatterers are moving, these non-stationary scatterers can cause a Doppler shift in the frequency. With an HCT level smaller than 2%, indicating a low number of scatterers, the scattered power becomes directly proportional to the density of scatterers. In the opposite situation, an increasing HCT level creates a decreasing averaged distance between the scatterers. As the decreased distance makes the scatterers' movement correlated, it results in a profound effect on the scattered power. [5]

The scattered power for a scattering volume of discrete scatterers or spatially fluctuating acoustic properties is expected to vary in terms of the incident intensity and scattering volume. A value describing the scattering strength of distribution of scatterers when hit by a plane harmonic wave is the differential scattering coefficient ϵ_d. Assuming that the element volume is of significant size in order to be a representative distribution of scatterers, the differential scattering cross-section is defined as the time-averaged scattered power $\bar{P}_s(\theta, \phi)$

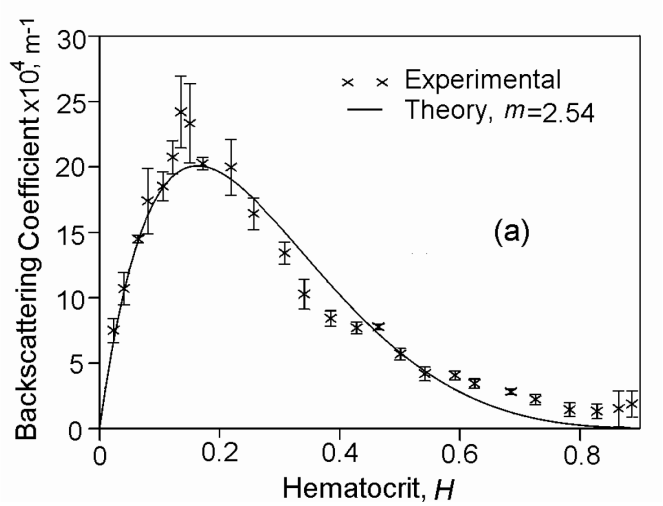


Figure 2.3: Comparison of experimental measurements for the backscattering coefficient ϵ versus hematocrit H with theoretical predictions. Reprinted from *Foundations of Biomedical Ultrasound* by Cobbold. [5]

in the direction (θ, ϕ) per unit solid angle divided by the product of the time-averaged incident intensity $\bar{I}$ and the element volume of scatterers dV. The coefficient is given in (2.3.7), and is a measure of the scattering strength of a distribution of scatterers when hit by a plane harmonic wave. [5] Furthermore, the backscattering coefficient ϵ can be defined in (2.3.8), as the differential scattering coefficient of 180 degrees, and as the backscattered cross-section per volume.

$$\epsilon_d(\theta, \phi) = \frac{\bar{P}_s(\theta, \phi)}{\bar{I} \cdot dV} \tag{2.3.7}$$

$$\epsilon = \epsilon_d(\pi, 0) = \frac{\sigma_b}{dV} \tag{2.3.8}$$

2.3.2 *Scattering of Bubbles*

With the Rayleigh approximation neglecting resonance, a model to consider for the scattering of bubbles is the linear oscillator. For doing so, the oscillation amplitude must be small relative to the equilibrium radius of the bubble. An analogy to the mechanical oscillator can be useful when modeling the bubble as a linear oscillator. The spring in the mechanical oscillator can be drawn a parallel to the gas pressure inside the bubble, the mass of a mechanical oscillator to the mass of the surrounding liquid being displaced, and the friction of a mechanical oscillator to the radiation resistance of the bubble.

The scattering of bubbles can be modeled as a harmonic oscillator. Hoff viewed the bubble as a harmonic oscillator in his report on *Acoustic Characterization of Contrast Agents for Medical Ultrasound*

Imaging, and a greater part of the theory deducted in this section is obtained from his work. [9]

The harmonic oscillator representation of the bubble can explain how oscillating bubbles get resonance frequency traits under linear conditions. Bubbles of small sizes in the ultrasound field will start to vibrate. By the bubble radius r, the resonance frequency f_r can be determined

$$f_r = \frac{1}{r}\sqrt{\frac{3\gamma\bar{p}_i}{\rho_0}}, \tag{2.3.9}$$

where γ is the specific heat of the gas, $\bar{i}$ is the average pressure inside the bubble, and ρ_0 is the density of blood. As the wavelength of the ultrasound signal is larger than the bubble radius, the bubble's surrounding pressure will oscillate around a mean value defined by the resonance radius r.

Oscillation is often regarded as a adiabatic process, meaning that there is no heat transport, when considering the compression of gas in acoustics. This is not the case for all bubbles of all sizes and frequencies. For bubbles of sizes of micrometers and frequencies at Megahertz, the oscillation is closer to isothermal. Hence, modelling the compression as a polytropic process can be of bigger convenience, as it describes the expansion and compression process, including heat transfer. For an ideal gas, the polytropic process is given as

$$pV^\kappa = \text{constant} \quad \Longleftrightarrow \quad \frac{dp}{p} = -\kappa\frac{dV}{V}, \tag{2.3.10}$$

where κ is the polytropic index and varies according to the thermodynamic process. [10] The polytropic index is $\kappa = 1$ for the isothermal process, and $\kappa = \gamma$ for isentropic processes. The adiabatic constant γ is the $\gamma_{air} = 1.40$ for air. The bulk modulus K can be given by the polytropic index

$$K = \kappa p_e, \tag{2.3.11}$$

where p_e is the equilibrium pressure.

When the bubble oscillates, there are minor volume changes ΔV around an equilibrium volume V_0. The momentarly radius $a(t)$ can be through the equilibrium radius a_e and the radial displacement $\xi(t)$.

$$a(t) = a_e + \xi(t), \quad |\xi| \ll a_e \tag{2.3.12}$$

The radial displacement is much smaller than the radius of the bubble, and can be used to express the change in volume

$$\Delta V = 4\pi a^2 \xi, \tag{2.3.13}$$

where the bubble's volume is $V = \frac{4}{3}\pi a^2$. With a change in volume, comes a change in the pressure inside the bubble. With the polytropic expression given in Equation (2.3.10), the pressure can be given as

$$\Delta p = p - p_e \quad \Longleftrightarrow \quad \Delta p = -\frac{3\kappa p_e}{a}\xi \tag{2.3.14}$$

Pressure is defined as force per area. Hence, the force can be given as

$$F_s = \int\int_S p_s \, dS = 4\pi a^2 p_s = -12\pi a \kappa p_e \xi = -s\xi, \tag{2.3.15}$$

where $s = 12a\kappa p_e$ is the spring constant. Thus, the bubble compression follows Hooke's law.

When the bubble oscillates, the surrounding liquid is set in motion, and inertia is introduced to the system. There is also interia added from the mass of the gas in the bubble. However, as the interisa introduced by the surrounding liquid is much larger in comparison, this addition of interia can be neglected. For the calculation of the inertia, the pressure field p_s radiation from the oscillating bubble can be used. The pressure field is a diverging spherical wave with a radial variation and is given by the pressure $p_s(a)$ at the surface of the bubble.

$$p_s(r) = p_s(a)\frac{a}{r}e^{i(\omega t - kr)}. \tag{2.3.16}$$

The relation between the velocity u of the surrounding liquid and the pressure p is given by the Euler equation with the non-linear term

$$\rho\frac{\partial u}{\partial t} = -\Delta p. \tag{2.3.17}$$

At the bubble surface, $u \to \dot{\xi}$ and $r \to a$, and Equation (2.3.17) becomes

$$\rho\dot{\xi} = -\frac{\partial p_s}{\partial r} = \frac{p_s(a)}{a}(1 + ika). \tag{2.3.18}$$

As the pressure p_s radiated from the bubble is given as

$$p_s(a) = \rho a \frac{1 - ika}{1 + (ka)^2} i\omega\dot{\xi} \approx \omega\rho a(ka + i)\dot{xi}, \quad ka \ll 1, \tag{2.3.19}$$

the force F_m on the bubble surface from the liquid motion can be calculated as in Equation (2.3.20). By expressing the force with the mechanical impedance Z_m, the force F_m is given by the radiation resistance R and the radiation reactance ωm in Equation (2.3.21).

$$F_m = -4\pi a^2 p_s = -4\pi a^3 \rho\omega(ka + i)\dot{xi} \tag{2.3.20}$$

$$F_m = -Z_m\dot{xi} = -(R\dot{\xi} + m\dot{\dot{\xi}}) = -(R + i\omega m)\dot{\xi}. \tag{2.3.21}$$

From Equation (2.3.20) and (2.3.21), the effective mass m of the oscillating bubble is given as

$$m = 4\pi a^3 \rho, \tag{2.3.22}$$

equaling three times the mass of the bubble at $\frac{4}{3}\pi a^3 \rho$.

There are multiple causes to the damping of the oscillations of the bubble. The main causes are the radiation resistance, viscosity in the surrounding liquid, and heat transport between the gas and surrounding liquid. The frictional force causing the damping is given in Equation (2.3.23). The mechanical resistance R of the oscillating bubble equals the sum of the three mechanisms producing the damping $R = R_c + R_\eta + R_{Th}$.

$$F_R = -R\dot{\xi} \tag{2.3.23}$$

Radiation introduces dampening of the oscillating bubble because the bubble loses energy as sound energy is radiated. The radiation resistance R_c is given by the impedance Z_m in Equation (2.3.24).

$$R_c = \mathrm{Re}(Z_m) = 4\pi a^2 \rho c (ka)^2 \tag{2.3.24}$$

Due to viscosity in the surrounding liquid, the oscillating bubble of micrometers experience a mechanical resistance. The viscosity is calculated from the viscous stress-tensor $T_L = -p_L - 4\eta\frac{\dot{a}}{a}$, which is the radial stress at the bubble surface, and where η is the shear viscosity. The radial stress introduces a viscous force F_η on the bubble surface, displayed in Equation (2.3.25). Together with Equation (2.3.23), the viscous resistance can be recognized.

$$F_\eta = -4\pi a^2 4\eta\frac{\dot{\xi}}{a} = -16\pi a\eta\dot{\xi} = -R_\eta\dot{\xi} \quad \Longleftrightarrow \quad R_\eta = 16\pi a\eta \tag{2.3.25}$$

The remaining cause of damping of the oscillating bubble is the heat transfer between gas in the bubble and the surrounding liquid. Assuming a constant temperature of the surrounding liquid, the thermal resistance is given in Equation (2.3.26), with Φ being the velocity potential, defined by $u = \Delta\Phi$.

$$R_{Th} = \frac{12\pi a p_e}{\omega}\,\mathrm{Im}\left(\frac{1}{\Phi(a,\omega)}\right) \tag{2.3.26}$$

By balancing the forces at the bubble surface, the equation of motion is found. The forces include the inertia from the surrounding liquid F_m, the damping from frictional forces F_R, the stiffness from the gas F_s, and the driving acoustic pressure $p_i(t)$. The equation of motion is qiven in Equation (2.3.27).

$$F_m + F_R + F_S = \int\int_S p_i(t)\,dS \tag{2.3.27}$$

As the expressions of the forces are inserted, the differential equation for the radial displacement ξ is given in Equation (2.3.28). The frequency domain is found by taking the Fourier transform, and is given in Equation (2.3.29).

$$m\ddot{\xi} + R\dot{\xi} + s\xi = -4\pi a^2 p_i \tag{2.3.28}$$

$$(-\omega^2 + i\omega\omega_r\delta + \omega_r^2)\hat{\xi}(\omega = -\frac{1}{\rho a}\hat{p}_i(\omega) \tag{2.3.29}$$

The angular resonance frequency ω_r is given by $\omega_r = \frac{s}{m} = \frac{3\kappa p_e}{a^2\rho}$. The damping constant δ is given by the angular resonance frequency as $\delta = \frac{R}{\omega_r m}$, and equals the sum of the different factors for damping $\delta = \delta_c + \delta_\eta + \delta_{Th}$. From the Equaion (2.3.29), the expression for the radial displacement is given as

$$\hat{\xi}(\omega) = \frac{1}{\rho a \omega_r^2}\frac{\hat{p}_i(\omega)}{(\omega/\omega_r)^2 - 1 - i\omega/\omega_r\delta}. \tag{2.3.30}$$

The sound pressure p_s radiated at the surface of the bubble in the frequency domain is found by taking the Fourier transform of Equation (2.3.32), as shown in (2.3.31). Combining the (2.3.32) and (2.3.31), the relation between the radiated pressure $\hat{p}_s$ and incident pressure is given in (**??**).

$$\hat{p}_s(a,\omega) = a\rho\ddot{\hat{\xi}}\frac{1}{1+ika} \approx -\omega^2 a\rho\hat{x}i(\omega) \tag{2.3.31}$$

$$\hat{p}_s(a,\omega) = \frac{\Omega^2}{1-\Omega^2+i\omega\delta}p_i(\omega), \quad \Omega = \frac{\omega}{\omega_r}, \tag{2.3.32}$$

Ω is the normalized frequency.

With the scattering cross section being the scattered power per incident sound intensity, the scattering cross section can be expressed by the ratio between the pressure amplitudes of the incoming and scattered field.

$$\sigma_s(a,\omega) = 4\pi a^2\left|\frac{p_s(a,\omega)}{p_i(\omega)}\right|^2 \tag{2.3.33}$$

$$\begin{aligned}\sigma_s(d,\omega) &= \pi d^2\frac{\Omega^4}{(1-\Omega^2)^2+(\Omega\delta)^2},\\ \Omega &= \frac{\omega}{\omega_r}, \quad \omega_r = \frac{s}{m} = \frac{3\kappa p_e}{(\frac{d}{2})^2\rho}.\end{aligned} \tag{2.3.34}$$

The relation between the two pressure amplitudes gives the scattering cross-section, displayed in (2.3.34), where $d = 2a$ is the diameter of the bubble.

2.3.3 *Emboli-to-Blood Ratio*

Information about emboli can be found from the Doppler US signal as the air bubble pass through the part of the blood vessel inside the ultrasound beam. Information about the volume of the emboli can be estimated based on the ratio between the peak amplitude of the Doppler signal from the emboli to the amplitude of the background blood Doppler signal, which often is estimated from parts of the signal with no emboli present. The ratio is defined as the emboli-to-blood ratio (EBR). [17] The EBR ratio can be estimated through the isonation of a blood vessel with an US wave of a single frequency, and if the HCT-level and blood vessel diameter is known.

From the pressure p at a distance D from the transducer probe and the acoustic impedance Z_0 of the medium the wave is propagating through, the acoustic incident intensity I_D can be defined.

$$I_D = \frac{p^2}{Z_0} = \frac{p^2}{\rho \cdot c} \tag{2.3.35}$$

By the use of (2.3.35), the acoustic intensity of an air bubble with a diameter of $d_{bubble} = 100$ µm that is at a distance $D = 3$ cm from the transducer probe can be calculated.

With an acoustic pressure of $p = 100$ kPa at distance D from the probe, the acoustic incident intensity at the bubble's position is $I_D = 6493.51$ W m^{-2}. The calculated acoustic incident intensity is based on the approximation that the density of blood is close to the density of water $\rho_{blood} \approx \rho_{water} = 1000$ kg m^{-3}, and that the mean propagation speed of sound in human tissue is 1540 m s^{-1}. [1]

With the assumption that shape of a bubble is a geometrically perfect sphere, the scattering cross section of a bubble with radius r_{bubble} is given in (2.3.36). With a radius of $r_{bubble} = d_{bubble}/2 = 50$ µm, the scattering cross section becomes $\sigma_{bubble} = 7.85 \cdot 10^{-9}$ m^2.

$$\sigma_{bubble} = \pi r_{bubble}^2. \tag{2.3.36}$$

With the scattering cross section being defined as the time-averaged scattering power in backward direction per time-averaged incident intensity, the total acoustic power P_{bubble} from the bubble can be expressed as in (2.3.37), and found to be $P_{bubble} = 51.0$ µW. [5]

$$P_{bubble} = I_D \cdot \sigma_{bubble} \tag{2.3.37}$$

Based on the assumption of a perfect geometric sphere, the acoustic power will scatter with an even distribution along a larger sphere. By dividing the total acoustic power P_{bubble} over an area σ_D of a sphere of radius equal to the distance from the probe, $r_D = D = 3$cm, the acoustic intensity of the reflected signal can be estimated to be $I_{bubble} = 4.51 \cdot 10^{-3}$W m^{-2}.

$$I_{bubble} = \frac{P_{bubble}}{\sigma_D} = \frac{P_{bubble}}{4\pi r_D^2}. \tag{2.3.38}$$

In a similar manner, the acoustic intensity of the reflected signal from blood at the same distance D from the transducer probe can be calculated. For a blood vessel of radius $r = 0.85$ mm at an angle of $\theta = 45°$relative to the transducer probe, the sample volume shaped as a cylindrical disk with elliptical flat ends and height $h = \frac{c}{2}\frac{N_p}{f_0}$ can be considered. The elliptical flat end will have a semiminor axis equal to the radius r of the blood vessels, while the semimajor axis will be equal to $r/\cos\theta$. From the expressions of the semimajor and semiminor axis, the area $A_{ellipse}$ of the ellipses can be calculated as shown in (2.3.39). With a frequency of $f_0 = 7.8$MHz, and a number of samples $N_p = 10$, an expression for the sample volume is given in (2.3.40).

$$A_{ellipse} = \pi\frac{r^2}{\cos\theta}. \tag{2.3.39}$$

$$dV = A_{ellipse} \cdot h = \pi\frac{r^2}{\cos\theta} \cdot \frac{c}{2}\frac{N_p}{f_0} \tag{2.3.40}$$

Then, the sample volume equals $3.17 \cdot 10^{-9}$ m^2. As the blood vessel is located at a distance D from the transducer, the acoustic incident intensity at the blood vessel is I_D, and the pressure is p.

Assuming normal hematocrit levels in the blood, the HCT level is set equal to 40. The backscattering coefficient ϵ is defined by the total backscattering intensity, incident intensity and elementary volume of scatterers. With HCT$=$ 40, the backscattering coefficient equals $\epsilon_{1(HCT)} = 7.0 \cdot 10^{-4}$ m^{-1}, obtained at a frequency of $f_1 = 7.5$ MHz. Assuming Rayleigh scattering of the RBC, the backscattering coefficient is given by (2.3.41).

$$\epsilon = \epsilon_{1(HCT)} \cdot \frac{f_0^4}{f_1^4} \tag{2.3.41}$$

$$P_{blood} = I_D \cdot \epsilon \cdot dV \tag{2.3.42}$$

Thus, the total acoustic power of the volume sample of the blood vessel can be calculated, with the use of (2.3.42), to be $P_{blood} = 1.69 \cdot 10^{-8}$ W. The intensity of the reflected signal from the blood vessel can then be calculated through (2.3.43)

$$I_{blood} = \frac{P_{blood}}{\sigma_D}, \tag{2.3.43}$$

to be $I_{blood} = 1.49 \cdot 10^{-6}$ W m^{-2}.

With the intensity of the signal from the bubble I_{bubble} being equal to $4.51 \cdot 10^{-3}$W m^{-2}, and the intensity of the signal from the blood I_{blood} in the blood vessel inside the ultrasound beam being equal

to $1.49 \cdot 10^{-6}$ W m^{-2}, EBR can be calculated in dBs as presented in (2.3.44). The resulting EBR is equal to 34.8 dB.

$$\text{EBR} = 10 \cdot \log_{10} \frac{I_{\text{bubble}}}{I_{\text{blood}}} \tag{2.3.44}$$

In reality, the observed signal of the bubble will equal the total of the ultrasound signal from the bubble, blood and background noise. Consequently, consideration of the measured emboli-to-blood ratio MEBR might be more convenient instead of the EBR. An expression for the MEBR is given in (2.3.45). [4]

$$\text{MEBR} = 10 \cdot \log_{10} \frac{I_{\text{bubble}} + I_{\text{blood}}}{I_{\text{blood}}}. \tag{2.3.45}$$

The approximation was used by Chung et al. in their paper on *Size Distribution of Air Bubbles Entering the Brain During Cardiac Surgery*, and later by Lam in her paper on *Estimation of Air Bubbles in The Cerebral Circulation in Newborns By Ultrasound Doppler Technique*, and might be closer to an actual received ultrasound signal from blood with an air bubble. Although it might be a closer approximation, it is not an exact representation of the actual ultrasound signal. In Figure 2.4, the estimation of the relation between the bubble diameter and the MEBR from (2.3.45) is displayed.

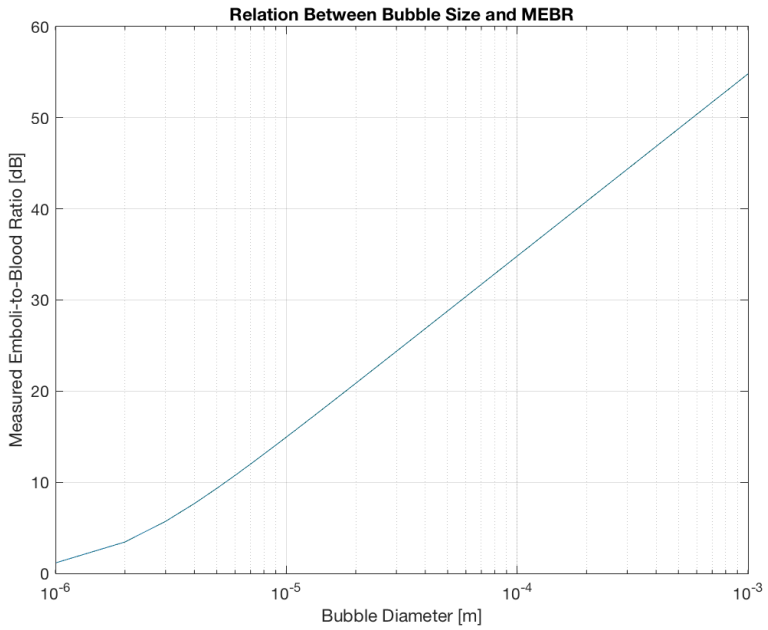


Figure 2.4: Relation between bubble diameter and measured emboli-to-blood ratio (MEBR), here the measured embolic signal is the signal from both the emboli and the blood. The figure is reprinted from Lam's paper on *Estimation of Air Bubbles in The Cerebral Circulation in Newborns By Ultrasound Doppler Technique*. [12]

With an MEBR approximation, the signal taken into account will be the sum of the signal from the bubble and blood, as well as any noise present in the signal. The signal-to-noise ratio (SNR) describes the ratio between the blood signal and the noise signal and is defined as the ratio between the power signal of the two, as shown in (2.3.46).

$$\mathrm{SNR} = 10 \cdot \log_{10} \frac{P_S}{P_N} \tag{2.3.46}$$

The blood signal consist of white noise of a root mean square (RMS) amplitude of 0 dB, equaling an amplitude of $\mathrm{rms}_{\mathrm{noise}} = 1$. With SNR = 12 dB, the amplitude of blood is equal to $\mathrm{rms}_{\mathrm{blood}} = 10^{\frac{12}{20}}$. The expected RMS amplitude of the signal including both the blood and the noise is given in (2.3.47).

$$\mathrm{rms}_b = \sqrt{\mathrm{rms}_{\mathrm{noise}}^2 + \mathrm{rms}_{\mathrm{blood}}^2}. \tag{2.3.47}$$

Denoting the IQ signal from blood and noise as IQ_b, and the IQ signal from the bubble as IQ_e, the measured signal from the bubble is assumed to be the maximum value of the envelope of the total signal, which is the maximum of the magnitude of the sum of the signals as shown in (2.3.48). [12]

$$\begin{aligned} |IQ_b + IQ_e|^2 &= (IQ_b^* + IQ_e^*) \cdot (IQ_b + IQ_e) \\ &= |IQ_b|^2 + |IQ_e|^2 + 2 \cdot \mathrm{re}(IQ_b^* + IQ_e), \end{aligned} \tag{2.3.48}$$

In Figure **??**, an example of the envelope of such an IQ signal containing a single bubble. Through (2.3.49),MEBR can be estimated and is a representation of what is actually being measured when estimating MEBR from the received US signal of blood containing an air bubble. $\hat{P}_b$ is the mean value of the squared IQ signal of the time positions without the bubble.

$$\mathrm{MEBR} = 10 \cdot \log_{10} \frac{\max(|IQ_b + IQ_e|^2)}{\hat{P}_b}, \tag{2.3.49}$$

Predicting the maximum value of (2.3.48) is not simple since the IQ signal IQ_b of the blood is a Gaussian signal. Thus, the first and third terms of (2.3.48) can easily get sporadically large values. However, the maximum value of (2.3.48) will approach $|IQ_e|^2$ for large values of EBR.

Part III

ALGORTIHM

3 ALGORITHM FOR THE DETECTION OF EMBOLI

The complete algorithm for the detection of emboli is in the appendix. The algorithm was developed in Matlab and consisted of several different variables, functions, and loops. The outline of the algorithm developed in this project is displayed in Figure 3.1.

Starting the algorithm is the loading of the ultrasound recording that is desired to analyze. Information from the record is stored in the struct `rec` for easy handling in the coming functions. From the recording, further information about the ultrasound signal is extracted and saved in another struct called `sig`. This struct includes the power signal, time and depth axis, and size of sample volume. Furthermore, the power signal is low pass filtered, and the Doppler shift is calculated. Following this step is a function for locating time positions with artifacts in the ultrasound signal that will be excluded from the detection function, which is the next step. The function for detecting embolic signals is done for one depth at a time, and at the beginning of each detection, an estimation of the background signal is calculated through the function `estimateBgs()`. Since the detections are performed depth by depth, and the bubbles signals usually distend over several depths, a function to correct for duplicate detections is performed. A function called `changeFormat()` is performed in order to get the detections in the same format as the detections done by the clinicians for easy comparison. Furthermore, the overview of the manual detections is gathered from the excel files. All the detections of a ratio above the chosen threshold are extracted for comparison with the detections done by the algorithm. The comparison is performed by the `compare()` function. Finally, the results are stored in excel-files.

3.1 LOW-PASS FILTER

The extracted power signal from the recording is low-pass filtered early on in the algorithm before any analysis is performed. The purpose of this step is to remove any fast fluctuations of the signal. Consequences of this process include minimizing the impact of false positives. The filtering removes the fast fluctuations of the signal, which can cause the algorithm to detect false positives. However, the filtering also cause a small reduction the amplitude of the peaks of bubble signals. On the contrary, as the filtering eases the fast fluctuations in the signal, it also minimizes the chance of a signal of a single bubble being recognized as several bubbles. The filtering also helped achieve a more precise locating of artifact positions.

Listing 1: Low-Pass Filtering of the Power Signal

```
prf    = 1/sig.tIncr;
[b,a]  = butter(2,fc/(prf/2));     % Butterworth filter of 2nd
    ↪ order
sig.pow_dB = filter(b,a,sig.pow_dB,[],2);
```

3.2 SEARCH OF TIME POSITIONS WITH ARTIFACTS

There are several types of artifacts that can affect an ultrasound image. The image in Figure 3.2 displays the Color M-Mode image of an ultrasound signal with artifacts and bubbles. The artifacts can be recognized at the time positions around 431 seconds and 434-435 seconds. Compared to the bubbles that can be found around 432-433 seconds, this type of artifact is easy to discriminate from the intensity increase of the signal due to the bubble.

Diathermy can typically introduce artifacts in the ultrasound image, causing straight lines down along all of the depths of the CMD. Motion artifacts can also cause artifacts in the uppermost depths. Probe movement can cause artifacts, especially in the shallower depths. X-ray and other electronic equipment can also cause mild continuous artifacts. Because of artifacts existing in the recordings, the algorithm must have a method for distinguishing between if peaks in the signal are actual bubble signals or just artifacts.

In order to avoid motion artifacts causing false positives in the algorithm, a decision was made to exclude the first eight sample volumes from the top of the recording from being included in the analysis of the algorithm. This decision was made because the motion artifacts usually only appear in the uppermost depths of the ultrasound image. Furthermore, the eight first sample volumes correspond to about the first 10 mm of the recording. Usually, there are no veins or arteries this close to the surface. In addition, there has not been observed any emboli in these depths of all the recordings of the training of the algorithm.

Throughout the development process of the algorithm, several different methods for detecting possible artifacts in the ultrasound signal were tested out. Some methods involved distinguishing between artifacts and bubbles after performing general detections for all areas of the signal above a certain threshold, while other methods involved detecting the artifacts before detection and excluding these areas when running the detection function.

One possible way of detecting the distinguishing between artifacts and bubbles was to introduce a demand for the peaks to hold a diagonal manner in the Color M-mode image. With the bubbles being in the blood vessels, there is expected a movement of the bubble that will cause it to appear as a diagonal line in the Color M-mode, as opposed to the artifacts, that will have lines in the Color M-mode of

a more straight nature. As this algorithm's detection is based on the amplitude of the signal, this would mean looking for peaks that are close to each other in time positions but not precisely at the same time position in depths right after each other to find the diagonals. This method was used for categorizing the detections made by the algorithm after the detection function. All time positions in each depth with an amplitude above the chosen threshold are detected in the detection function. Then, the artifact check would happen in the next part of the algorithm, where the algorithm goes through all the detections performed and categorize them as either an artifact or bubble. The categorization was based on how close detections in different depths were to each other, and if they were close enough to might be considered as the detections of the same bubble, it would remove the detection with the lowest amplitude of the two in order to correct the amount of counted bubbles. If the detections were even closer in time, to be categorized as lying straight after one another, the detection would be categorized as an artifact, and both detections would be removed. This method turned out to be challenging to implement in a manner that gave good results. Having too strict demands for diagonality caused some of the bubbles with straighter-looking signals to go undetected, and having less strict demands caused many artifacts to be classified as bubbles. It also caused the algorithm to mistake artifacts for bubbles and bubbles for artifacts, giving an overall wrong detection.

Another attempt was to break down the categorization part into two steps. Among the detections of embolic signals, the first step was to search for artifacts, then check for duplicates of bubbles. Firstly, the received detections were checked to see if they were close to each other in both depth and time. If they were, the mean of their time positions would be calculated and compared to the subsequent detection that would lie close to these two detections in time in other depths. If that detection also lay close in time, it would be added to the calculation of the meantime position of the current artifacts, added to a count, and so on until it went empty of close detections. If the count of the detections from different depths lying at approximately the same time positions then was above a certain number of detections, for example, 10, they would be classified as an artifact, and they would all be removed from the list of detections to be further checked for as duplicates detections of already detected bubbles. However, this method did not give excellent results. Even though the artifacts would show straight lines across all depths in the CMD, not all of the correspondings peaks of the amplitude would necessary be above the threshold, causing not all depths of the artifact to get a detection from the detections function. Thus, when looking for detections close to each other, they might be further apart, causing the algorithm to classify them as bubbles wrongly.

Then, we tried doing an artifact search at the beginning of the algorithm, instead of simultaneously as the bubble detection and correction, as mentioned earlier. This method was based on calculating the mean of the power signal from each depth. The primary purpose of doing so is to catch the prominent artifacts throughout all or most of the depths. By taking the mean of the power signal over the depths, time positions with artifacts will have peaks in most of the depths, and the resulting mean power will as well. While for bubble signals, there are only peaks in the power signal in some depths, and the mean power signal will not have as prominent peaks as for the artifacts. Setting a certain threshold, all time positions where the amplitude of the mean power signal is above the threshold will be saved. Further down the algorithm, when it is time for the detection function, these time positions will not be included for evaluation.

Lastly, the function estimate the variables `startInd` and `endInd`, which will be stored in the struct `sig` and used for setting what time position further the algorithm should start and stop the analysis. For recordings with, for example, zero artifacts, the variables `startInd` and `endInd` will be the first and last time position of the recording. However, if there are artifacts at the beginning or end of the recording, like situations where the ultrasound probe might be turned on after the recording has started, the analysis will start after the artifact at the beginning, shorting down the running time of the algorithm.

Listing 2: Function for Locating Time Positions with Artifacts

```
function [art, sig] = findArtLoc(sig, rec, threshArt, N_art)
    fileName = rec.fileName;
    pow_dB   = sig.pow_dB;
    t        = sig.t;

    meanPow  = mean(pow_dB,1);
    meanmean = 4;
    artLoc1  = meanPow>(movmedian(meanPow,N_art)*(1+threshArt));
    artLoc2  = meanPow<meanmean;
    artLoc   = zeros(1,length(meanPow));

    for i = 1:length(meanPow)
       if (artLoc1(i)==1) || (artLoc2(i)==1)
           artLoc(i) = 1;
       end
    end

    sig.startInd = 1;
    while (artLoc(sig.startInd)==1)
        if (sig.startInd == length(artLoc))
            break
        end
        sig.startInd = sig.startInd + 1;
    end

```

```
    sig.endInd = length(artLoc);
    while (artLoc(sig.endInd)==1)
        if (sig.endInd == 1)
            break
        end
        sig.endInd = sig.endInd - 1;
    end

    artTime0     = artLoc .* t;
    art          = struct();
    art.artTime  = artTime0(artTime0~=0);
    art.artCount = length(art.artTime);
end
```

3.3 ESTIMATION OF THE BACKGROUND SIGNAL

The method for detecting embolic signals in the algorithm is based on finding peaks of high amplitude where the amplitude is of a certain amount of decibel (dB) above the background power signal, being the Doppler signal of blood. For doing so, the background signal must first be calculated. This is done through the function `estimateBgs()`. In the estimation of the background signal, any peaks in the power signal must be excluded from the calculation to avoid getting a too high background signal. This is done by calculating a threshold where all peaks above shall be excluded from the estimation. The threshold was calculated as the moving median of the power signal times 140%. The time positions with amplitudes above the threshold were removed from the calculation and replaced with values using the inbuilt Matlab function `fillmissing()`. Finally, the background signal was calculated using a moving mean method for each time position of the power signal by using the N number of points before and after the current time position.

The function is run for each depth inside the function `detectEmb()`. However, the function only estimates the background signal if the power signal is not classified as pulsative. In depths with pulsatile blood flow, the amplitude of the power signal, whether there are bubbles present or not, will have amplitudes of a wide range of decibel (dB). This is shown in Figure 3.3, the difference between a normal and pulsatile blood flow. This estimates the background signal non-representative and further causes a lot of false-positive detections. In order to avoid this situation, the standard deviation of the power signal is calculated. All depths where the power signal has a standard deviation above a certain value, the estimated background signal will be assigned the value `NaN`. When moving forward in the algorithm, detection of the power signal in a depth will only be initiated if the background signal is not equal to `NaN`.

Listing 3: Function for Estimation of The Background Power Signal

```
function [bgs_n,counts,centers] = estimateBgs(rec,sig,pow_n,n,N)
    fileName = rec.fileName;
    t = sig.t; zIncr = sig.zIncr;

    % Exclude inadequate depths from being included in detection
    nbins = 10;
    [counts, centers] = hist(pow_n, nbins);
    highSpread = (sum(counts>(sum(counts)*0.05))) >= (nbins/2);

    % Check if highSpread:
    if highSpread
        bgs_n = NaN;
    else
        % Exclude start and end points from being used in
            ↪ estimation of bgs
        pow_fix = pow_n;
        iStart  = max([sig.startInd,30]); % 30 timepoints * tIncr
            ↪  = 0.1 s
        iEnd    = min([sig.endInd,(length(t)-30)]);

        for i = 1:length(pow_n)
            if (i<iStart) || (i>iEnd)
                pow_fix(i) = NaN;
            end
        end
        pow_fix = fillmissing(pow_fix,'movmean',N);

        % Exclude peaks from estimation of bgs
        bgs_pos = pow_fix < (movmedian(pow_fix,N)*1.4);
        hollow_bgs = pow_fix;

        for i = 1:(length(pow_fix)-1)
            if bgs_pos(i)==0
                hollow_bgs(i)=NaN;
            end
        end

        filled_bgs = fillmissing(hollow_bgs, 'movmean', N);
        bgs_n = movmean(filled_bgs, N);
    end
end
```

In this function, the power signal is analyzed for embolic signals. At the beginning of the function, several variables are created, ready to store the results from the search. The detection function uses a single-gated method for finding the embolic signals, meaning that the search process happens at one depth at a time. However, the detection function will search all depths, and duplicate detections of the same bubble will later be corrected in the algorithm, making the whole algorithm a multi-gated algorithm. From earlier in the algorithm, in the function `artLoc()`, two variables `startInd` and `endInd`

were found. Based on what is the maximum of `startInd` and 30 time positions (roughly equals 0.1 seconds), the search for embolic signals begin at this point, and similarly, based on what is the minimum of `endInd` and 30 time positions from the end, the search ends at this point for each depth. For each depth, the `estimateBgs()`-function is called, and the background signal for that depth is estimated. As earlier mentioned, if the `estimateBgs()`-function finds that the power signal has an amplitude range of high spread, like for example is of pulsatile blood flow, the estimation is set equal to `NaN`. Before beginning the search, the detection function will check if the estimation of the background signal was set to `NaN`. If so, the function will not search for embolic signals in this depth and move on to the next depth. However, if the estimation is not equal to `NaN`, the process of detection will carry on. The function iterates through the time points of the signal's amplitude in that depth. If the time difference between the current time point being analyzed and the closest time position with a detected artifact is below a chosen variable `artWidth`, the algorithm will move on to the next time position. If not, the analysis will move on to check if this time point has an amplitude above the threshold, which is the sum of the estimated background signal and chosen EBR in dB. If so, it will save this time position and continue to check for how long the amplitude stays above the threshold and if the distance to the closest artifact is above a certain `artWidth`. Simultaneously, it will also update a variable called `bubLength` to keep track of the bubble length in time. When either the amplitude of the threshold drops below the threshold or the distance to the closest artifact becomes smaller than the chosen `artWidth`, the algorithm will check if the bubble length is above the expected length of a bubble signal, calculated from the Doppler shift. If so, the highest amplitude of the time interval analyzed will be stored in a list called `bubSig_n` at the corresponding time point, and a variable keeping track of the number of detections, called `embCount` will be iterated by one. The script will continue this method throughout the time points of the amplitude of the signal from the current depth until it reaches the `endInd`.

When the analysis of one depth is finished, the count of bubbles is added to an overall count for the whole recording, called `embCount_all`. The list `bubSig_n` is also stored in an overall list for the recording, called `bubSig_all`, at the index matching the depth it represents. The same is done for the estimated background signal of the current depth, `bgs_n` in the overall list, called `bgs_all`. From the list `bubSig_n`, three other lists for the depth is also created, `timeIdx_n` with the time of the different detections in seconds, `bubAmp_n` with the amplitudes of all the detections, and `Ibub_n` with the index of all the detections in time points. All of these three are also stored in an overall list correspondingly. All of the above-mentioned overall lists are stored in the struct `res` for further analysis.

Listing 4: Function for the Detection of Bubble Signals

```
function [res] = detectEmb(sig,rec,art,EBRmin,n0,N,artWidth)
    pow_dB = sig.pow_dB; nd = sig.nd;
    tIncr = sig.tIncr;   t = sig.t;
    zIncr = sig.zIncr;   T = sig.T;
    startInd = sig.startInd; endInd = sig.endInd;
    artTime = art.artTime;

    res = struct();     % Save results in struct

    bubSig_all = {};   bgs_all = {};
    thresh_all = {};   embCount_all = 0;
    timeIdx_all = {};  bubAmp_all = {};
    Ibub_all = {};

    iStart = max([startInd,30]); % 30 timepoints * tIncr = 0.1 s
    iEnd   = min([endInd,(length(t)-30)]);
    for n = n0:nd
        pow_n = pow_dB(n,:);
        [bgs_n,counts,centers] = estimate_bgs(rec,sig,pow_n,n,N);
        bgs_all{n} = bgs_n;
        thresh = bgs_n + EBRmin;
        thresh_all{n} = thresh;
        bub_sig_n = NaN([1 length(pow_n)]);
        bub = 0;
        emb_count = 0;

        if (~isnan(bgs_n)) % Ensures the signal from depth n is
            ↪ adequate
            j = 1;
            for i=iStart:iEnd
                ti = i*tIncr;
                [minVal, closestIndex] = min(abs(artTime-ti));
                closestArtTime = artTime(closestIndex);
                expectedLength = round(T(n,i)/tIncr);
                bubLength = 0;
                if abs(closestArtTime - ti) > artWidth
                    if (i>=j) && (pow_n(i)>thresh(i)) &&(j<=iEnd)
                       j = i;
                       bubLength = 1;
                       tj = ti;
                       while (pow_n(j)>thresh(j)) &&
                           ↪ (abs(closestArtTime-tj)>artWidth)
                           ↪ && (j<iEnd)
                           bubLength = bubLength + 1;
                           j = j + 1;
                           tj = j*tIncr;
                           [minVal,closestIndex]=min(abs(artTime-
                               ↪ tj));
                           closestArtTime = artTime(closestIndex)
                               ↪ ;
                       end
```

```
                    if bubLength > expectedLength
                        bub = bub + 1;
                        [val, idx] = max(pow_n(i:j));
                        idx = idx + (i-1);
                        bub_sig_n(idx) = val;
                        emb_count = emb_count + 1;
                    end
                end
            end
        end

        embCount_all  = embCount_all + emb_count;
        bubSig_all{n} = bub_sig_n;

        Ibub          = find(~isnan(bubSig_all{n}));
        Ibub_all{n}   = Ibub;
        timeIdx       = Ibub * tIncr;
        timeIdx_all{n} = timeIdx;
        bubAmp        = bub_sig_n(~isnan(bub_sig_n));
        bubAmp_all{n}  = bubAmp;
    end
    res.bubSig_all   = bubSig_all;
    res.bgs_all      = bgs_all;
    res.thresh_all   = thresh_all;
    res.embCount_all = embCount_all;
    res.timeIdx_all  = timeIdx_all;
    res.bubAmp_all   = bubAmp_all;
    res.Ibub_all     = Ibub_all;
end
```

3.4 FUNCTION FOR THE CORRECTION OF DUPLICATE DETECTIONS

Because this is a multi-gated algorithm, there is a need for a function to correct for detections of the same bubble appearing in several depths. That is the purpose of the function `correctDuplicates()`.

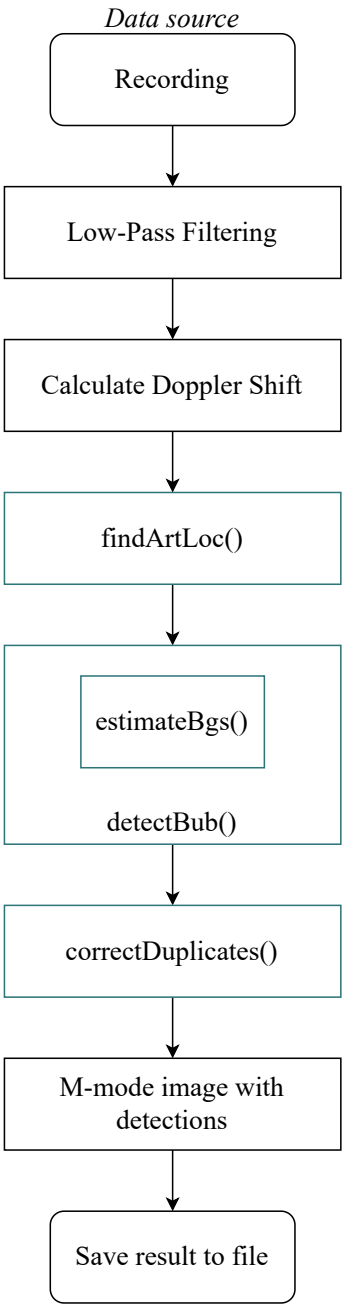
The functions iterate through all the detections made in `detectEmb` through a double for-loop, making sure to be able to compare each detection with each other. The function checks how far apart the detections are in both time and depth for each pair of detections. It checks if the two detections are close diagonal distance, further diagonal distance, or very close horizontal distance. There are set different criteria for these. If they are, the detection with the highest amplitude will be kept, and the other will be removed. Then, the algorithm continues in the same manner until it has iterated through all the detections previously made. The struct `res` is updated with the lists.

Listing 5: Function for the Correction of Duplicate Detections

```
function res = correctDuplicates(sig,res,tMinD,tMaxD,tMinN,zMin,
    ↪ zMax)
```

```
timeIdx_all  = res.timeIdx_all;
bubAmp_all   = res.bubAmp_all;
bubCount_all = res.embCount_all;

nd = sig.nd; zIncr = sig.zIncr; tIncr = sig.tIncr;

for n1 = 1:nd
    tBub1 = timeIdx_all{n1};
    for n2 = 1:nd
        tBub2 = timeIdx_all{n2};
        for i = 1:length(tBub1)
            for j = 1:length(tBub2)
                tDiff = abs(tBub1(i)-tBub2(j));
                zDiff = abs(n1*zIncr-n2*zIncr);
                bub1exists = ~isnan(timeIdx_all{n1}(i));
                bub2exists = ~isnan(timeIdx_all{n2}(j));

                % remove duplicates of same bubble in close
                    ↪ diagonal direction
                if (n1~=n2) && (zDiff<zMin) &&
                    ↪ (tDiff<tMinD) && (bub2exists) &&
                    ↪ (bub1exists)
                    if (bubAmp_all{n1}(i)<bubAmp_all{n2}(j))
                        timeIdx_all{n1}(i) = NaN;
                        bubAmp_all{n1}(i)  = NaN;
                        bubCount_all = bubCount_all - 1;
                    else
                        timeIdx_all{n2}(j) = NaN;
                        bubAmp_all{n2}(j)  = NaN;
                        bubCount_all = bubCount_all - 1;
                    end

                % remove duplicates of same bubbles in
                    ↪ further diagonal distance
                elseif (n1~=n2) && (zDiff>=zMin) &&
                    ↪ (zDiff<zMax) && (tDiff>tMinD) &&
                    ↪ (tDiff<tMaxD) && (bub2exists) &&
                    ↪ (bub1exists)
                    if (bubAmp_all{n1}(i)<bubAmp_all{n2}(j))
                        timeIdx_all{n1}(i) = NaN;
                        bubAmp_all{n1}(i)  = NaN;
                        bubCount_all = bubCount_all - 1;
                    else
                        timeIdx_all{n2}(j) = NaN;
                        bubAmp_all{n2}(j)  = NaN;
                        bubCount_all = bubCount_all - 1;
                    end

                % remove duplicates of same bubble in
                    ↪ horizontal direction
                elseif (n1==n2) && (i~=j) && (tDiff<tMinN) &&
                    ↪  (bub2exists) && (bub1exists)
```

```
                        if (bubAmp_all{n1}(i)<bubAmp_all{n2}(j))
                            timeIdx_all{n1}(i) = NaN;
                            bubAmp_all{n1}(i)  = NaN;
                            bubCount_all = bubCount_all - 1;
                        else
                            timeIdx_all{n2}(j) = NaN;
                            bubAmp_all{n2}(j)  = NaN;
                            bubCount_all = bubCount_all - 1;
                        end
                    end
                end
            end
        end
    end
    res.timeIdx_all  = timeIdx_all;
    res.bubAmp_all   = bubAmp_all;
    res.bubCount_all = bubCount_all;
    res.tMinD = tMinD;
    res.tMaxD = tMaxD;
    res.tMinN = tMinN;
    res.zMin  = zMin;
    res.zMax  = zMax;
end
```



Load the power signal [dB] from the recording

Decrease the effect of fast fluctuations in the power signal.

Calculate the Doppler shift in order to calculate the expected length of a bubble signal.

Check the recording for artifacts, and locate these time positions. These will be excluded from further analysis.

Search the power signal for air bubbles depth by depth. Before performing detection, the background signal is estimated through the estimateBgs() function for each depth.

Correct the number of detections, remove duplicate detections of the same bubble.

Plot the final detections in the Color M-Mode.

Figure 3.1: Diagram of the algorithm for automatic detection of air bubbles.

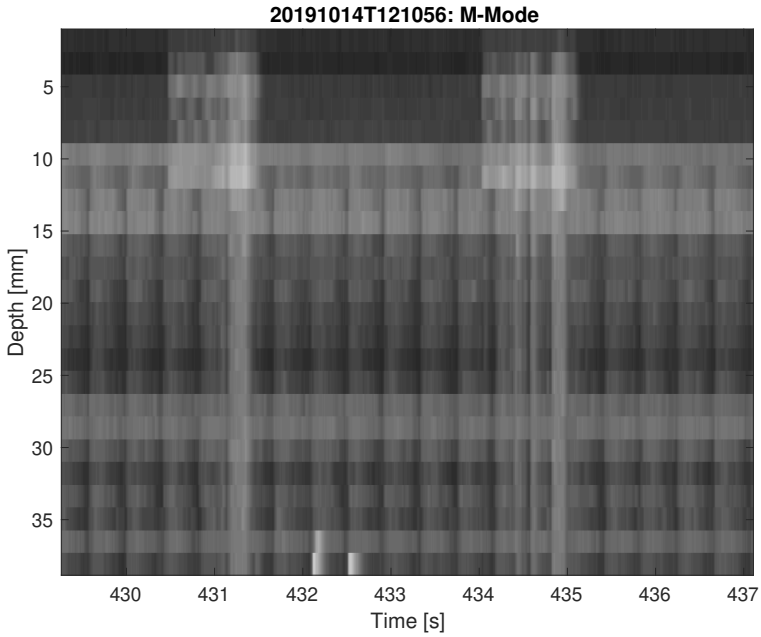
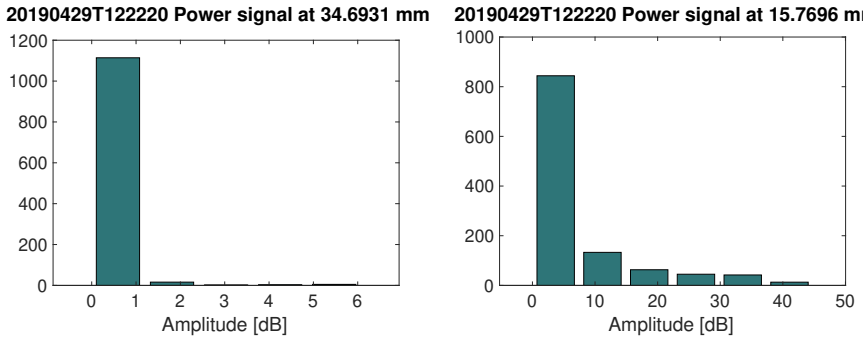


Figure 3.2: Color M-Mode (CMD) of a ultrasound signal containing artifacts and bubble. The artifacts are located at around 431 s and 435 s, while the bubbles are located between 432 and 433 s.



(a) (b)

Figure 3.3: Histogram of the amplitudes of a power signal with normal blood flow in 3.3a and pulsatile blood flow in 3.3b.

Part IV

METHODS

In this chapter, the methods used in this project will be presented. The chapter starts with a description of the recordings and patients included in this project. Furthermore, the ultrasound composition and equipment will be described. Finally, the chapter finishes with the description of the algorithm developed.

4 METHODS

In this chapter, the methods used in this project will be presented. The chapter starts with a description of the recordings and patients included in this project. Furthermore, the ultrasound composition and equipment will be described. Finally, the chapter finishes with a description of the algorithm developed.

4.1 PATIENTS AND RECORDINGS

In this project, actual US recordings have been included for both the training and testing of the algorithm. The recordings included were taken at Oslo University Hospital (OUS). The recordings included are gathered from Leth-Olsen et al. from their study on *Detection of Cerebral High Intensity Transient Signals by NeoDoppler During Cardiac Catheterization and Cardiac Surgery in Infants*. The study was a collaboration between the Department of Pediatric Cardiology at Oslo University Hospital (OUS) in Oslo and the ultrasound group at the Norwegian University of Science and Technology (NTNU). The 31 patients included in the study were neonates and infants of less than one year of age with CHD who were scheduled for transcatheter intervention or cardiac surgery with cardiopulmonary bypass (CPB).

In total, 28 of the patients from the study have been included in this project. Three of the patients from Leth-Olsen et al.'s study, patients 1, 6, and 18 from the catheter intervention, were excluded from the results because of various difficulties and disruptions with the US recordings. Therefore, these three patients are also excluded from this project. Of the 28 patients included, 15 (54%) of the patients had catheter interventions, and 13 (46%) of the patients had cardiac surgery with CPB. The patients had a median age of 3.2 months, ranging from 0.1 to 8 months old.

There were 650 recordings in total. The length of the recordings ranges from 60 seconds to 30 minutes, depending on the type of intervention and the purpose of the intervention.

4.2 ULTRASOUND COMPOSITION

4.2.1 *NeoDoppler Ultrasound System*

The ultrasound system used in this project is called the NeoDoppler and is a non-invasive ultrasound Doppler system which is an US technology for continuous measurements of cerebral blood flow in

neonates or newborns. The system is developed by the Ultrasound Group at the Department of Circulation and Medical Imaging at NTNU in Trondheim, Norway. The ultrasound system consists of a small, lightweight probe, an ultrasound scanner, and a PC with a user interface. The ultrasound probe has a diameter of 10 mm and operates at a frequency of 7.8 MHz. The pulse is transmitted at a rate of 8 kHz, as it emits plane waves that cover a cylindrical shape of depths up to 38 mm and a diameter equal to the diameter of the probe. The sample volumes are 1.5770 mm in depth and 0.0025 s in width.

The equipment aims to reduce brain injury in premature infants and critically ill neonates by monitoring cerebral blood flow, which is performed through the open fontanelles infants have. By placing the probe on top of the fontanelles, it is possible to measure the blood flow continuously. The ultrasound system consists of the ultrasound probe, the ultrasound module with a power supply, and a computer with software for processing and displaying the data. The software is further described in the next section.

4.2.2 *EarlyBird Software*

The software EarlyBird was developed for the NeoDoppler Ultrasound system and is intended to be used for the non-invasive measuring of cerebral circulation. The system presents the raw data from the ultrasound transducer probe and scanner as both a Color M-Mode image and a Doppler spectrum that can be analyzed. The system includes several adjustable parameters, such as the gain and the dynamic range, which opens up for dynamic filtering, making it possible to search bubbles and emboli in the blood flow at multiple depths simultaneously. The multi-gated system is helpful for the recognition of bubbles that appear at different depths at different times and or that stretch in time or depth. The image of the software is displayed in Figure 4.1.

4.3 MANUAL DETECTIONS

In order to evaluate whether the algorithm is making accurate detections or not, data on manual detections are included in the project. For all the recordings included in this project, which are the same as Leth-Olsen et al. used in their study, Leth-Olsen et al. has performed manual detections of detected air bubbles in the recordings. These will be used in the algorithm to compare with the detections the algorithm gets. The manual detections, also called high intensity transient signals (HITS), were based on an "embolic signature" in the Color M-Mode (CMD) with corresponding high intensity in the Doppler spectrogram. The detection process was performed after the intervention

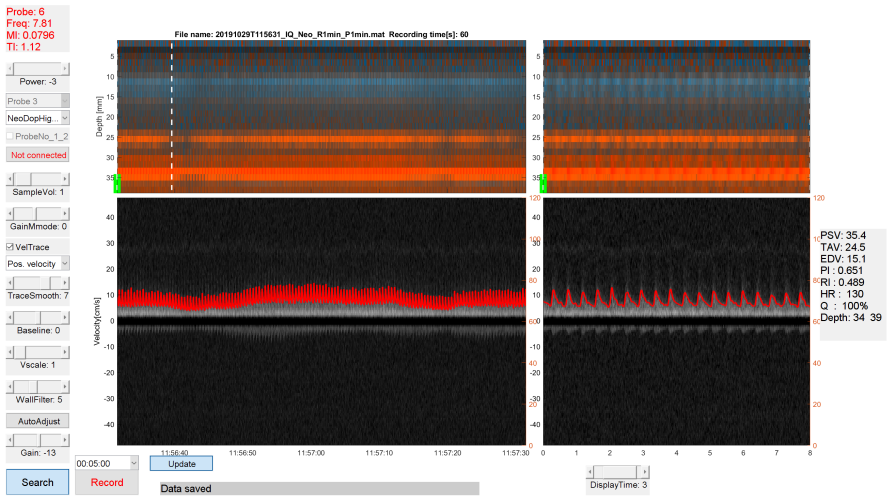


Figure 4.1: EarlyBird Software

and with the use of EarlyBird Software and MatLab. The recordings are visualized in Earlybird with CMD and Doppler spectrogram.

Together with the use of the script called `bubCount_Manual.m`, written by Hans Torp, the HITS were detected manually in EarlyBird, and further information about the detections was calculated from the script. The script is included in the appendix. For each detection marked manually in EarlyBird CMD, the script saves the time and depth position of the HITS. Then, a small rectangular area around the manual detection will be analyzed. The highest amplitude within this area will be stored, and the background level is estimated from the power signal of the same depth by taking the mean of the time positions of the power signal where the power signal is lower than the median of the power signal plus two times the standard deviation of the power signal. Then, the EBR of the HITS can easily be calculated by subtracting the background level from the amplitude in dB. The resulting format on the data on the manual detections performed by Leth-Olsen et al. are displayed in Figure 4.2.

SingleBubles	tb	zb	AmpdB	Amplitude	Background_dB	Ratio_dB
2019:04:29:11:05:59:055	737544,4625	28,58887959	14,7	25,35010617	8,155452726	6,544547274
2019:04:29:11:05:59:610	737544,4625	19,15897702	21,3	32,60769216	7,689473684	13,61052632
2019:04:29:11:06:00:149	737544,4625	28,33401736	19,8	44,97964322	2,289092737	17,51090726
2019:04:29:11:06:00:149	737544,4625	36,48960878	29,1	203,6370727	10,50299476	18,59700524

Figure 4.2: Manual detections performed by Leth-Olsen et al.

4.4 THE ALGORITHM FOR THE DETECTION OF AIR EMBOLI

4.4.1 *Training the algorithm*

The algorithm was created in Matlab, based on similar principles of the algorithm Kjelsaas used for the algorithm developed in the study

on *Detection of Air Emboli in the Brain of Neonates by Ultrasound Doppler*. [11]

At the beginning of the development of the algorithm, the value of some parameters was chosen by conjecture. Hence, for the training of the completed algorithm, these parameters were further adjusted to find the combination of values of the parameters that gave the best result. These parameters are displayed in Table 1 with an associated description of the parameter.

The start depth (`n0`) was trained for `n0 = 7` and `n0 = 8`, equaling depths at 11-13 mm from the surface. These values were chosen for the training as there usually are no arteries above these depths, as well as the superior depths being more exposed to motion artifacts. The parameter `N` was trained for values between 100 to 4069 samples, corresponding to 0.25 - 10 seconds. The value $N_a$rtwastunedforvaluesbetween250to100
25seconds.Fortheparametercalledthreshart,thealgorithmwastrainedforvaluesbetw
60%.Fortheparametersartwidth, tMinD, tMaxDandtMinN,theyweretrainedforvaluesb
0.8seconds.TheparameterszMinandzMaxweretrainedforvaluesbetween3 −
13mm.Finally,thelastparameterstdLimwastrainedforvaluesof3to5dB.Thelastparam
OlsenEtal.intheirstudyon*Detection of Cerebral High Intensity Transient Signals by NeoDop*

4.4.2 *Comparison of The Results of The Algorithm with The Manual Detections Performed by Clinicians*

All the recordings included in this project have been collected by clinicians Martin Leth-Olsen and Siri-Ann Nyrnes from St. Olavs Hospital in Trondheim, Norway. In addition, they have also gone through all the recordings and manually detected the bubbles with time, depth, amplitude, and amplitude ratio in dB. By taking advantage of this, the algorithm can use this data to compare with after the detections of the algorithm are performed to evaluate the detection's performance. In Listing **??** is the function used to evaluate the performance of the detection.

In a similar manner to the `correctDuplicates()`-function, this function uses a double `for`-loop to iterate through all the automatic detections and all the manual detections. For each pair of automatic and manual detection, the algorithm calculates how far apart the detections are from each other in time and depth. First, all detections are compared looking at a close diagonal distance. Then, if any are still not matched, the algorithm compares within a larger diagonal distance, now with a demand that the distance also must be larger than a certain distance, to ensure the bubble detection is not straight. If the detections passes the criteria for being the same detection, the automatic detections are added to the lists called `tCorrect` and `zCorrect`, and the manual detections are added to the lists called `tChecked` and `zChecked`. They are also removed from the list containing the detec-

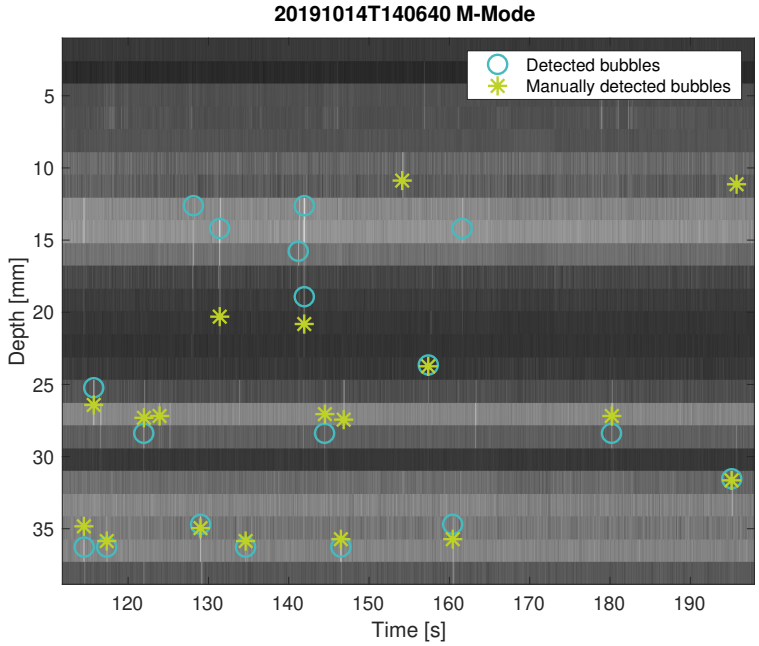


Figure 4.3: Color M-Mode image with the detections done by the algorithm (turquoise circles) compared to those manually counted (green stars).

tions up for analysis. The final results will be stored in a struct called `comp`.

The manual detections included in the algorithm are filtered based on the EBR of the detections. An inclusion threshold is calculated to be the chosen `EBRmin` value of the algorithm, including a margin of 30%. The margin is added because the calculation of the EBR is calculated in different methods and thus, is expected to differ in value. With an `EBRmin` equal to 15 dB, the inclusion threshold is equal to 10.5 dB.

4.4.3 *Optimal parameters*

In order to decide on which combination of parameters obtained the most promising result, a calculation was performed based on how many correct, false, and missed detections the different combinations achieved. For each set of parameters, the ratio of correct detections (`%correct`) was calculated by dividing the number of correct detections by the number of manual detections. The ratio for the missed detections (`%missed`) was calculated in a similar manner by dividing the number of missed detections by the total number of manual detections. The ratio of false detections (`%false`) was calculated by dividing the number of false detections by the total number of detections obtained by the algorithm. Lastly, the performance (`P`) was evaluated

by the difference of the ratios, `%correct - (%missed` + %false), was calculated for all the different sets of parameters.

From a clinical perspective, it is beneficial to obtain the smallest ratio of missed detections. Thus, the set of parameters was ranked by the lowest ratio of missed detections. Amongst the highest-ranked sets of parameters, the set of parameters with the highest value for performance (`P`) was chosen as the best performing set of parameters. These parameters will be further used in the test set.

VARIABLE	DESCRIPTION
no	Number of samples of depths from the top to start the detection at
N	Number of points to include in the moving mean calculation of the background signal
N_art	Number of points to include in the moving mean calculation of `findArtLoc()` function
treshArt	The percentage of the moving mean calculation of the `findArtLoc()` function to create the threshold
artWidth	Minimum distance in seconds a detection can have to an artifact detection
tMinDiag	The minimum distance two detections can have in seconds in order to be classified as the detection of the same bubble
tMaxDiag	The maximum distance two detections can have in seconds in order to be classified as the detection of the same bubble
tMinNorm	The minimum distance two detections can have in seconds in order to be classified as the detection of the same bubble
zMin	The minimum distance two detections can have in depth in mm in order to be classified as the detection of the same bubble
zMax	The maximum distance two detections can have in depth in mm in order to be classified as the detection of the same bubble
stdLim	Maximum standard deviation of the amplitudes of a power signal of one depth in dB
EBRmin	The minimum ratio between the peak and the background signal a detection must have in dB

Table 1: Variables used in the algorithm with associated descriptions.

Part V

RESULTS

5 RESULTS

All of the following results are color-coded. Turquoise color represents detections and calculations performed by the algorithm. Green color represents detections and calculations performed by clinicians, and is used in comparison for verification of correctness in detection and for a measure of performance.

5.1 TRAINING THE ALGORITHM

5.1.1 *Optimal parameters of the algorithm*

After the process of training the algorithm, the combination of the values of the parameters that were found to give the best performance was the combination displayed in Table 2. The full overview of all combinations tested out is in the appendix. By using this particular set of values for the different parameters, the algorithm has a cut-off frequency `fc` of 20.3 Hz. Further, the detection process will begin at the 8^{th} sample volume from the top, roughly equal to a depth at 12.6 mm. For the estimation of the background signal, `N` $= 500$ samples before and after the current sample was included in the calculation for that sample. 500 time samples equal about 1.23 seconds. The same number of samples, `N_art` $= 500$, was used in order to estimate a signal for locating artifacts. This signal is multiplied by 1+ `threshArt`$= 1.3 = 130\%$. Further, any time positions closer than 0.1 seconds to a detected artifact will not be included in the detection part of the algorithm. For the detections that are made, they will be checked for whether or not they are of the same bubble as any of the other detections. This analysis will be based on the criteria of the bubbles having a time distance shorter than `tMinDiag` $= 0.2$ seconds and a depth distance shorter than `zMin` $= 7.88$ mm. Another criteria that can give the classification as detections of the same bubble, is the time distance being between `tMinDiag` and `tMaxDiag` $= 0.6$ s simultaneously as the distance in depth is between `zMin` and `zMax` $= 18.14$ mm. The last possibility for two detections to possibly be classified as of the same bubble is if the time distance is smaller than `tMinNorm` $= 0.075$ seconds if the detections are at the same depth. Lastly, any depth where the power signal has a standard deviation larger than 4 dB will be excluded from the analysis.

VARIABLE	VALUE	COMMENT
fc	20.3 Hz	
no	8 samples	Equals 12.6 mm
N	500 samples	Equals 1.23 seconds
N_art	500 samples	Equals 1.23 seconds
threshArt	0.3 s	Equals 30 %
artWidth	0.1 s	
tMinDiag	0.2 s	
tMaxDiag	0.6 s	
tMinNorm	0.075 s	
zMin	5 samples	Equals 7.88 mm
zMax	11.5 samples	Equals 18.14 mm
stdLim	4 dB	

Table 2: The combination of the values of the different parameters used for the training set that gave the most promising performance.

5.1.2 *The Training Set*

For the training set, a total of 68 recordings were used. Two patients were included in the training set, one from catheter intervention (CI) and one from heart surgery (HS). 26 of the recordings are from patient 5 from CI, and the other 42 recordings were from patient 3 from HS.

In Table 3, an overview of the detections achieved when using the combination of parameters shown in 2, is displayed. The table displays the overview of how many bubbles were detected by the algorithm compared to the manually counted by clinicians, as well as displays the overview of correct, missed, and false detections. The overview is both displayed per patient and type of medical intervention and as a grand total.

From the recordings of the catheter intervention from patient 5, the algorithm detected a total of 22 bubbles, compared to the 62 detections manually counted by clinicians. Of the detections performed by the algorithm, 18 (81.8%) of the 22 detections were classified as correct, as they matched with 18 (29%) of the 61 manual detections. Consequently, 4 (18.2%) of the automatic detections were categorized as false, and 44 (71%) of the manual detections were missed.

For the heart surgery, with the recordings of patient 3, there were in total 53 detections counted by the algorithm. Of the automatic detections, 34 (63.2%) were correctly counted, while 19 (35.8%) of the detections did not match with any of the manually counted bubbles. Of the 68 manual detections, 34 (50%) was detected by the algorithm as well, whereas the other 34 (50%) were missed.

In total, the algorithm detected 75 bubbles in the training set, where 52 (69.3%) were correctly detected and 23 (30.6%) were false detections. Only 52 (40%) of the 130 manual detections were revealed, while 78 (60%) of the manual detections were missed.

TRAINING SET

PATIENT NR.	AUTO. DETECTED	MANU. DETECTED	CORRECT	MISSED	FALSE
CI: 5	22	62	18	44	4
HS: 3	53	68	34	34	19
Total	75	130	52	78	23

Table 3: The results of the detections of the test set. CI - Catheter intervention, HS - Heart surgery.

In Figure 5.1, the distribution of the EBR of the correct detections are presented. For these detections, the figure displays the distribution of the EBR calculated by the algorithm in turquoise and the EBR calculated by the clinicians in green.

The EBR distribution of the missed detections of the training set are displayed in Figure 5.2. As these bubbles are missed by the algorithm, there are only EBR values estimated by the clinicians.

The false detections and the associated EBR values of the training set are displayed in Figure 5.3. With no matching manual detections for the false detections, there are only EBR values calculated through the algorithm.

5.2 TESTING THE ALGORITHM

The results of the test set are shown in the following figures.

5.2.1 *Test set*

Of the 650 recordings included in this project, 582 of them were included in the test set. From CI, there were included 14 different patients and 247 recordings. A total of 370 recordings from 12 different patients were included from the HS.

Displayed in Table 4 is an overview of the detections achieved when running the algorithm on the test set. The table displays the overview of how many bubbles were detected by the algorithm compared to the manually counted by clinicians, as well as displays the overview of correct, missed, and false detections. The overview is both displayed per patient, per type of medical intervention, and as a grand total.

From the recordings of the CI, a total of 86 detections were obtained. Of these detections, 57 (66.3%) of the 86 detections were classified

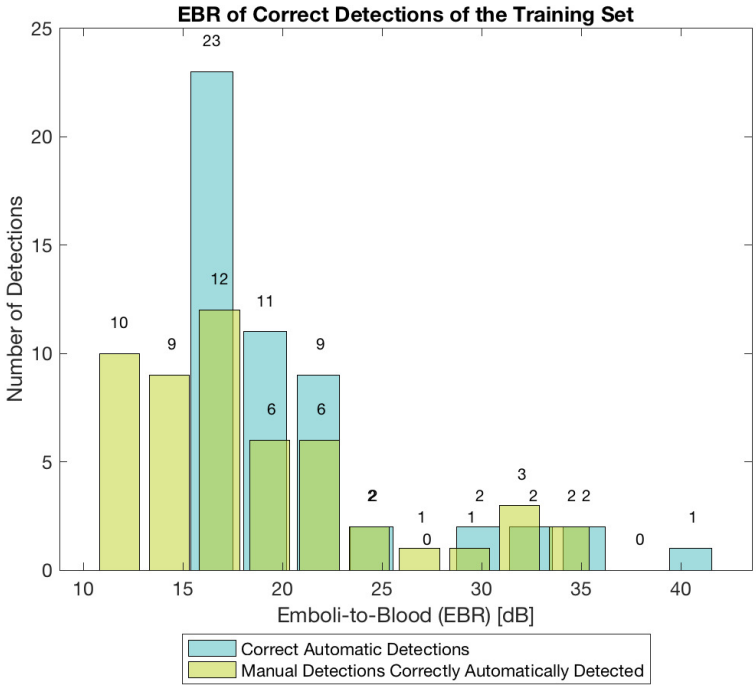


Figure 5.1: The emboli-to-blood ratio (EBR) in decibels (dB) of the correct detections of the test set.

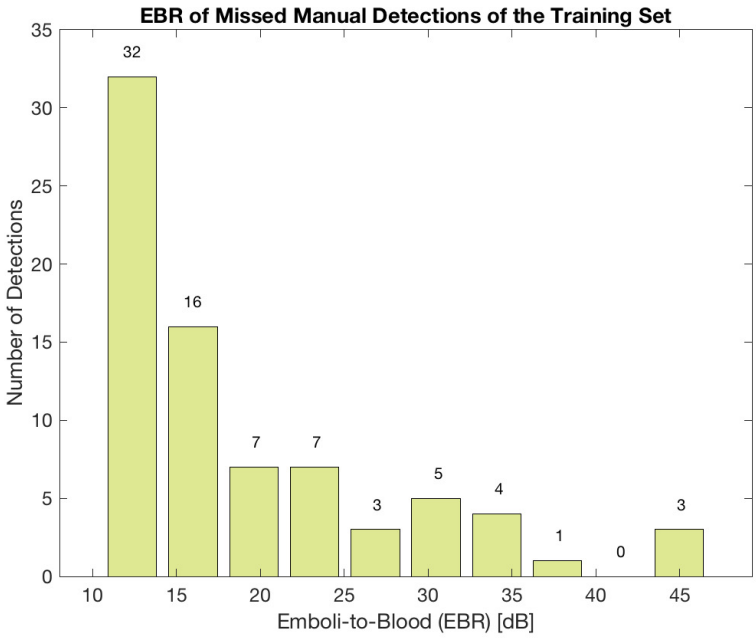


Figure 5.2: The emboli-to-blood ratio (EBR) in decibels (dB) of the missed manual detections of the test set.

as correct, as they matched with 57 (33.1%) of the 172 manual detections. Consequently, 29 (33.7%) of the automatic detections were categorized as false, and 115 (66.9%) of the manual detections were missed.

For the heart surgery, there were in total 265 detections counted by the algorithm. Of the automatic detections, 53 (20%) were correctly counted, while 212 (80%) of the detections did not match with any of the manually counted bubbles. Of the 197 manual detections, 53 (26.9%) was detected by the algorithm as well, whereas the other 144 (73.1%) were missed.

In total, the algorithm detected 351 bubbles in the test set, where 110 (31.3%) were correctly detected, and 241 (68.7%) were false detections. Only 110 (29.8%) of the 369 manual detections were revealed, while 259 (70.2%) of the manual detections were missed.

In Figure 5.4, the distribution of the EBR of the correct detections of the test set are presented. For these detections, the figure displays the distribution of the EBR calculated by the algorithm in turquoise and the EBR calculated by the clinicians in green.

The EBR distribution of the missed detections of the test set are displayed in Figure 5.5. As these bubbles are missed by the algorithm, there are only EBR values estimated by the clinicians.

The false detections and the associated EBR values of the test set are displayed in Figure 5.6. With no matching manual detections for the false detections, there are only EBR values calculated through the algorithm.

5.2.2 *Test set with all manual detections included*

For both the training and testing of the algorithm, there are only included manual detections of an EBR larger than the set inclusion threshold, which is equal to 10 dB for `EBRmin` equal to 15 dB. However, for analytic purposes, a comparison of the algorithm against the entire list of manual detections of the test set is included.

The results of the test set when comparing the detections made by the algorithm with all the manual detections are displayed in Table 5. The table displays the overview of how many bubbles were detected by the algorithm compared to all manual detections, as well as displays the overview of correct, missed, and false detections. The overview is both displayed per patient, per type of medical intervention, and as a grand total.

With all the manual detections included, 65 (75.6%) of the 86 detections were classified as correct, as they matched with 65 (23.5%) of the 277 manual detections. Thus, 21 (24.5%) of the automatic detections were categorized as false, and 212 (76.5%) of the manual detections were missed.

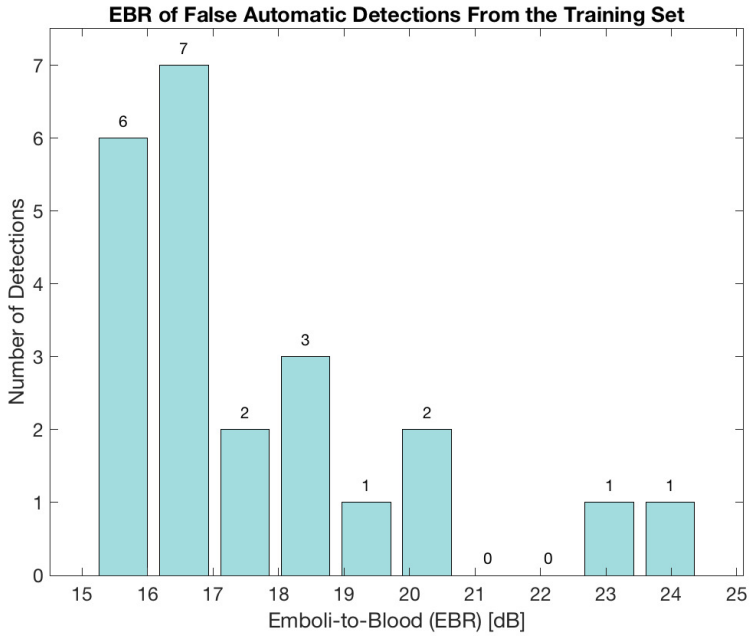


Figure 5.3: The emboli-to-blood ratio (EBR) in decibels (dB) of the false manual detections of the test set.

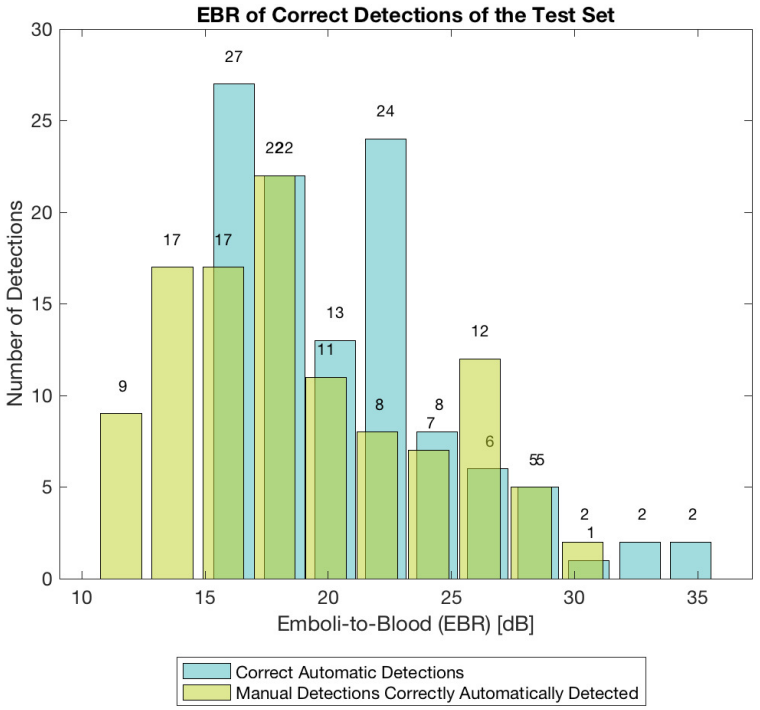


Figure 5.4: The emboli-to-blood ratio (EBR) in decibels (dB) of the correct detections of the test set.

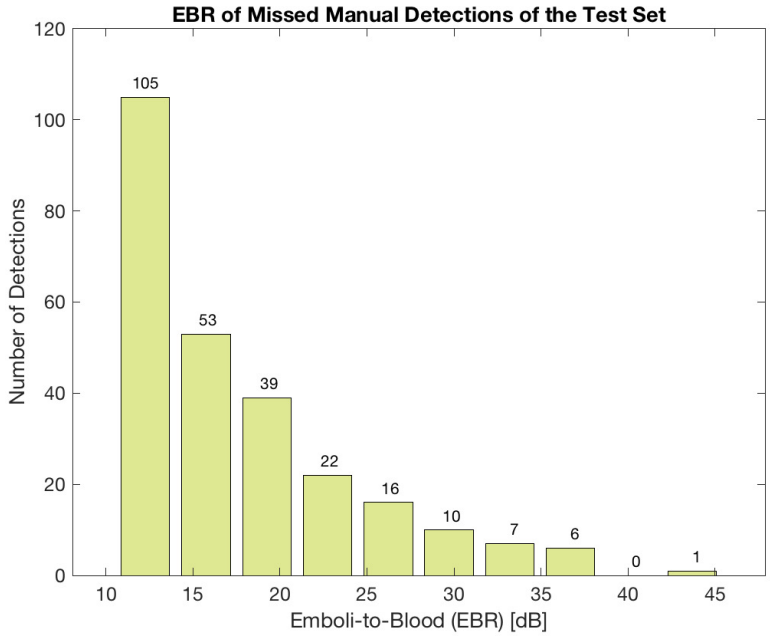


Figure 5.5: The emboli-to-blood ratio (EBR) in decibels (dB) of the missed manual detections of the test set.

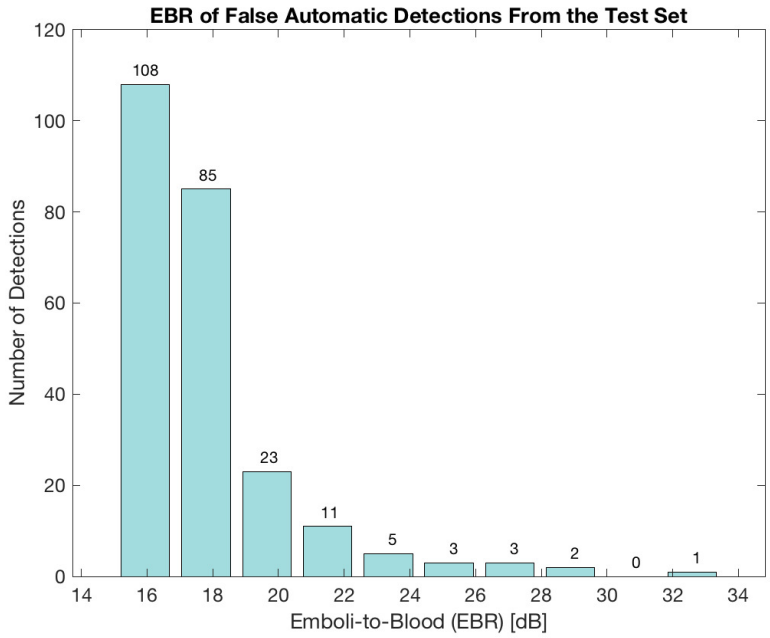


Figure 5.6: The emboli-to-blood ratio (EBR) in decibels (dB) of the false manual detections of the test set.

Including manual detections of no filtering, 56 (21.1%) of the detections from HS were correctly counted, while 209 (78.9%) of the detections did not match with any of the manually counted bubbles. Of the 578 manual detections, 56 (9.7%) were detected by the algorithm as well, whereas the other 522 (90.3%) were missed.

To sum up, with all manual detections included, the algorithm correctly detected 121 (34.5%) of 351 detections, while the other 230 (65.5%) were false detections. Only 121 (14.2%) of the 855 manual detections were revealed, while 734 (85.8%) of the manual detections were missed.

In Figure 5.7, the distribution of the EBR of the correct detections of the test set are presented. For these detections, the figure displays the distribution of the EBR calculated by the algorithm in turquoise, and the EBR of the manual detections that were detected by the algorithm and calculated by the clinicians in green.

The EBR distribution of the missed detections of the test set are displayed in Figure 5.8. As these bubbles are missed by the algorithm, there are only EBR values estimated by the clinicians.

The false detections and the associated EBR values of the test set are displayed in Figure 5.9. With no matching manual detections for the false detections, there are only EBR values calculated through the algorithm.

5.3 RUNNING TIME

The running time of the algorithm is presented in the section. For all recordings used for both the training set and test set, the time the algorithm uses to analyze the recording is saved. In addition, the duration of the recording is also saved, as longer recordings might cause the algorithm to use a longer time to analyze the recording.

Both the duration of the recordings and the average running time of the algorithm are displayed in Table 7 for the training set and in Table 8. In the tables, the average running time per 60 seconds is also given, along with the number of recordings used to calculate the average, giving the significance of the numbers. The calculation of the running time is per 60 seconds in order to make the average running time of the different duration of the recordings comparable.

As shown in Table 7, the average running time per 60 seconds for the training set was equal to 7.6 seconds per 60 seconds, equaling a running time of about 12.7 % of the length of the recording.

For the test set, however, the average running time per 60 seconds was equal to 3.3 seconds per 60 seconds, as shown in Table 8. Thus, the running time equals about 6% of the length of the recording.

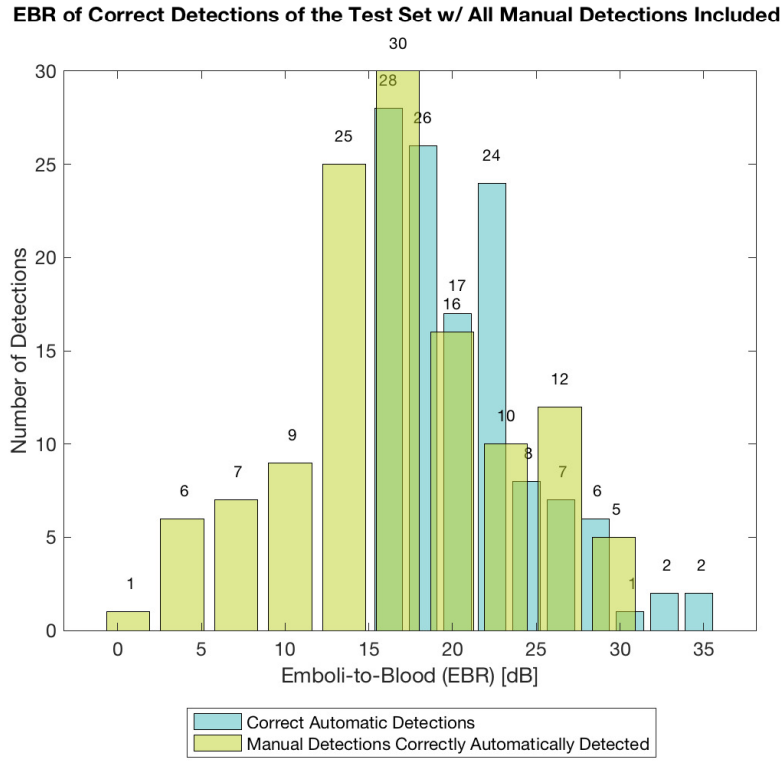


Figure 5.7: The emboli-to-blood ratio (EBR) in decibels (dB) of the correct detections of the test set when all of the manual detections are included.

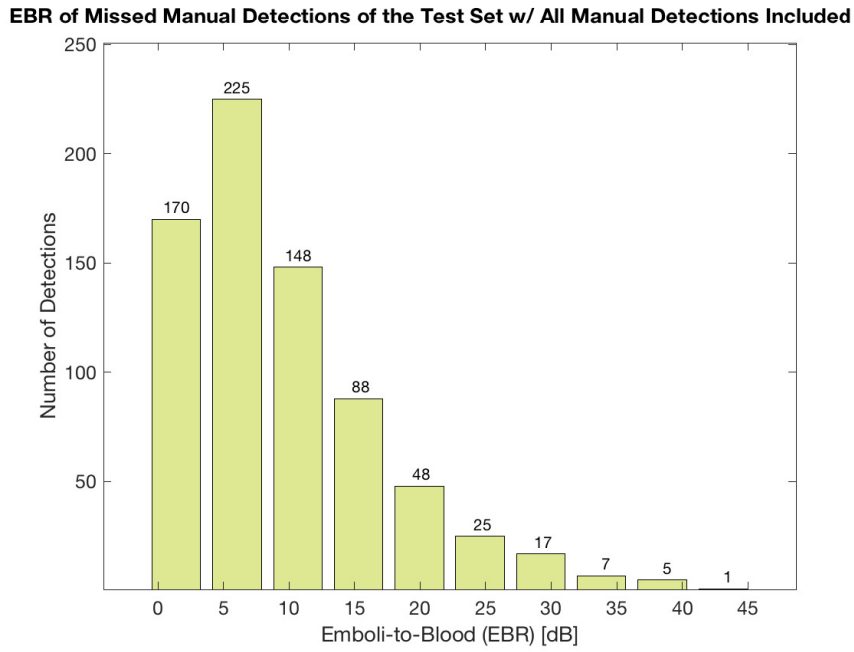


Figure 5.8: The emboli-to-blood ratio (EBR) in decibels (dB) of the missed manual detections of the test set when all of the manual detections are included.

TEST SET					
PATIENT NR.	AUTO. DETECTED	MANU. DETECTED	CORRECT	MISSED	FALSE
Catheter intervention					
2	3	6	3	3	0
3	9	12	4	8	5
4	2	11	2	9	0
7	3	6	0	6	3
8	0	1	0	1	0
9	0	6	0	6	0
10	1	3	0	3	1
11	0	0	0	0	0
12	0	0	0	0	0
13	10	17	8	9	2
14	0	0	0	0	0
15	55	89	38	51	17
16	2	18	1	17	1
17	1	3	1	2	0
Total	86	172	57	115	29
Heart surgery					
1	26	10	3	7	23
2	30	17	14	3	16
4	17	5	1	4	16
5	7	11	4	7	3
6	18	22	7	15	11
7	113	5	1	4	112
8	0	2	0	2	0
9	11	50	11	39	0
10	17	48	3	45	15
11	6	6	4	2	2
12	15	11	4	7	11
13	4	10	1	9	3
Total	265	197	53	144	212
Grand Total	351	369	110	259	241

Table 4: The results of the detections of the test set.

TEST SET WITH ALL MANUAL DETECTIONS

PATIENT NR.	AUTO. DETECTED	MANU. DETECTED	CORRECT	MISSED	FALSE
Catheter intervention					
2	3	7	3	3	0
3	9	29	4	8	5
4	2	13	2	9	0
7	3	12	0	6	3
8	0	1	0	1	0
9	0	9	0	6	0
10	1	7	0	3	1
11	0	0	0	0	0
12	0	0	0	0	0
13	10	24	8	16	2
14	0	0	0	0	0
15	55	148	46	102	9
16	2	19	1	18	1
17	1	8	1	7	0
Total	86	277	65	212	21
Heart surgery					
1	26	21	3	18	23
2	30	41	17	24	13
4	17	5	1	22	16
5	7	23	4	32	3
6	18	36	7	67	11
7	113	74	1	4	112
8	0	31	0	31	0
9	11	112	11	101	0
10	17	72	3	69	15
11	6	61	4	57	2
12	15	24	4	20	11
13	4	78	1	77	3
Total	265	578	56	522	209
Grand Total	351	855	121	734	230

Table 5: The results of the detections of the test set with all manual detections included.

	TEST SET						
	AUTO.	ALL MANU. DETECTED			MANU. DETECTED > 15 DB		
	TOT.	TOT.	% MISS	% FALSE	TOT.	% MISS	% FALSE
CI	86	277	76.5%	24.5%	172	66.9%	33.7%
HS	265	578	90%	78.9%	197	73.1%	80%
Tot	351	855	85.9%	65.5%	369	70.2%	68.7%

Table 6: Comparison of the result of the test set when including different amounts of the manual detections. Explanation of the abbrevations: CI = Catheter intervention, HS = Heart surgery, Auto. = Automatic detections, Manu. = Manual detections, Tot. = Total number of detections, %Miss = number of missed detections divided by total number of manual detections, %False = number of false detections divided by total number of automatic detections.

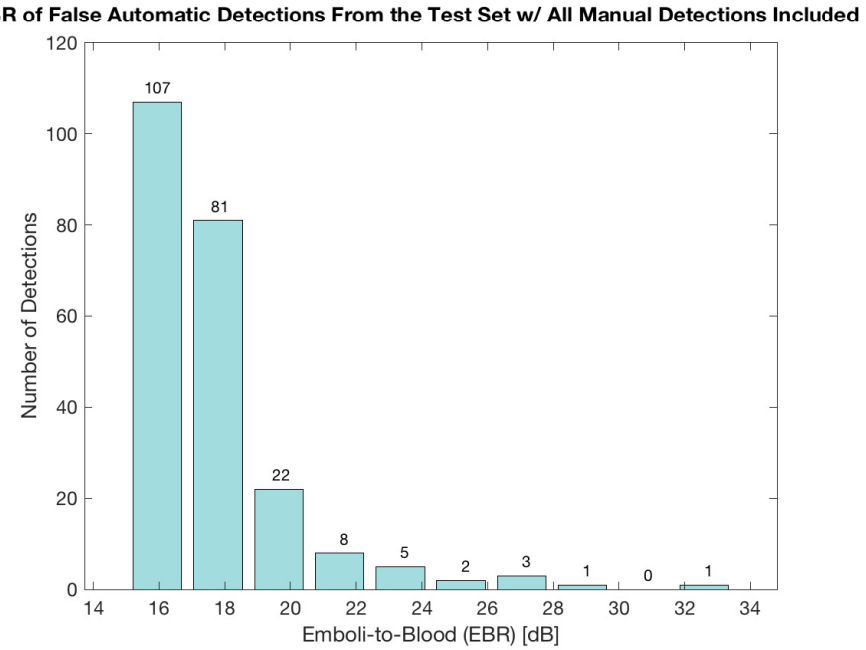


Figure 5.9: The emboli-to-blood ratio (EBR) in decibels (dB) of the false manual detections of the test set when all of the manual detections are included.

TRAINING SET

LENGTH OF RECORDING [s]	AVERAGE RUNNING TIME [s]	AVG. RUNNING TIME PER 60 s [s]	NUMBER OF RECORDING [#]
< 60	0.3	5.7	8
60	4.7	4.7	41
180	93.8	31.3	8
300	14.2	2.8	21
900	33.7	2.2	8
1340	29.2	1.3	1
1800	121.5	4.1	1
Total	20.4	7.6	68

Table 7: Average running time of the different lengths of the recordings from the training set.

TEST SET

LENGTH OF RECORDING [s]	AVERAGE RUNNING TIME [s]	AVG. RUNNING TIME PER 60 s [s]	NUMBER OF RECORDING [#]
< 60	1.3	5.6	80
60	2.2	2.2	383
60 - 300	21.1	9.8	29
300	14.2	2.8	21
300 - 900	165.2	13.7	6
900	69.6	4.6	38
900 - 1800	65.2	3.4	3
1800	72.9	2.4	22
Total	12.5	3.3	582

Table 8: Average running time of the different lengths of the recordings from the test set.

Part VI

DISCUSSION

6 DISCUSSION

6.1 THE TRAINING SET

From the training set, the algorithm detected a smaller number of detections than the manual detections. The results revealed that the algorithm performed a relatively exact detection rate, with 69.3% of all the automatic detections being correct, 81.8% for the CI and 63.2% for the HS. Thus, around 30% of the detections were categorized as false positives. However, when comparing the automatic detections with the manual detections, only 40% of the automatic detections in total were matched with the detections obtained by the clinicians. This results in a ratio where 60% of the detections are missed, which is not preferable from a medical point of view.

When looking at Figure 5.2, it is observed that a majority of the missed detections have an estimated EBR between 10 and 15 dB. Consequently, if there had not been added a margin for the set threshold of which manual detections that was included, and thus, only the manual detections with an EBR above 15 dB would have been included, the amount of missed detections would have been reduced from 78 (40%) to 46 (35.4%), as 32 of the missed detections were below 15 dB. However, observing the Figure 5.1, it would also reduce the number of correct detections, from 52 (69.3%) to 33 (44%) correct detections, because 19 of the manual detections the algorithm also detected had an estimated EBR calculated by the clinicians that were below 15 dB, which is the chosen threshold. Thus, it can be argued that the algorithm performs better with a margin at the inclusion threshold for the manual detections.

From Figure 5.3, it is observed that most of the EBR of the false detections are between 15 to 17 dB, having an EBR right above the threshold value. This suggests that these false detections might have barely reached the cut. Possibly, by training the algorithm on even more combinations of the value of the parameters, it might separate out these false detections.

The difference of the algorithm's performance between the recordings from the CI and HS are also of significance. Although the total number of manually counted detections is roughly close for the two patients, the number of detections performed by the algorithm is quite different. For the CI and patient 3, the number of detected bubbles is much smaller for the algorithm than the number of manual detections. However, the ratio of correct detections made by the algorithm is high. For patient 5 of HS, the opposite is true. The num-

ber of detections obtained by the algorithm is closer to the amount manually detected, yet, the ratio of false positives is larger. The difference in the recordings of the two patients might be too significant for the algorithm to achieve a solution that would give promising results for both of the categories. When training the algorithm, some of the recordings of patient 5 from CI containing most of the manual detections were difficult for the algorithm to detect without loosening up on the strict criteria of excluding possible areas with artifacts. If doing so, in order to be able to detect the bubbles of these recordings, it would cause more of the artifacts present in the recordings of patient 3 from HS to be wrongly detected as bubbles. Thus, with the best performance of the training set as an average of the performance of the two patients, it can have resulted in an intermediate solution for both of the cases. Consequently, caused some missed detections for patient 5 and simultaneously caused false detections for patient 3 in the form of artifacts.

6.2 THE TEST SET

For the test set, the algorithm detected almost as many bubbles as there are manual detections of the test set. However, the algorithm detected almost half the amount of manual detections for the patients of CI and detected a more considerable amount than the manual detections for the patients of HS.

The results of the test set show a large reduction in how exact the detection rate is, compared to the training set, with only 33.1% of all the automatic detections being correct. The numbers are 66.3% for the CI and 20% for the HS. Thus, around 68.7% of the detections were categorized as false positives. However, when comparing the automatic detections with the manual detections, only 29.8% of the automatic detections in total were matched with the detections obtained by the clinicians. This results in a ratio where 70.2% of the detections are missed, which is an even less preferable percentage compared to the training set.

Observing Figure 5.5, the distribution of the sizes of the EBR of the missing detections has a similar trend as for the training set. Even larger for the test set than the training set is the amount of missed detections with an estimated EBR below 15 dB. In fact, there are 105 detections with an EBR below 15 dB, around 40% of the total missed detections. Thus, changing the filtering on the included manual detections from 15 dB with a margin to a firm limit of 15 dB and above would reduce the amount of missed detections significantly, from 70.2% to around 42%. Again, changing the criteria for which manual detections will be included for comparison of performance would also introduce a reduction of the amount of correct automatic detections, from 33.1% to 24%. This is visible from Figure 5.4. However, the

reduction of correct automatic detections is small in comparison with the reduction of missed manual detections. Thus, a change of criteria could significantly improve the measurement of the performance of the algorithm on the test set.

From Figure 5.6, the distribution of EBR of the false detections is shown. Similar to the training set, a larger part of the detections have an EBR right above 15 dB. Further suggesting that these detections might have been filtered out if the resulting estimate for the background signal was calculated a bit differently. In addition, it indicates the importance of the estimation of the background signal for the detection process.

6.3 THE TEST SET INCLUDING ALL MANUAL DETECTIONS

When comparing the detections obtained by the algorithm to all of the manual detections, without any lower limit for the EBRs included, 11 of the detections that earlier were classified as false are now classified as correct. By comparing Figure 5.4 with Figure 5.7, it is observed that these 11 detections have an EBR value much lower than 15 dB when calculated for the manual detections. From the same figure, it is observed that these detections have EBR values above 15 dB when calculated by the algorithm. Thus, when only including manual detections above the inclusion threshold of 15 dB with a margin of 30%, some of the detections the algorithm performs are classified as false, even though they match with some of the manual detections, causing a higher percentage of false positives, as shown in Table 6. This is possibly caused by the fact that the method for estimating the EBR is different for the manual detections and the automatic detections, causing variations of the resulting EBR of each detection. With different methods of calculation, it is expected for the values of the different methods to differ. However, when the results differ as greatly as they do for this situation, it makes it more difficult to measure the performance of the algorithm based on the comparison with the manual detections. By having a filter on the manual detections included for comparison, the numbers of false detections and missed detections are less representative of the actual performance of the algorithm.

6.4 SOURCES OF ERROR

The algorithm had a significantly better performance for the training set than for the test set. This can be due to the training set being much smaller than the test set. With a testing set of only 10% of all the recordings included in the project, it might not be enough for the algorithm to be trained for all the different possible cases of ultrasound recordings, explaining why the algorithm performed significantly better for the training set. By having a larger training set, the

algorithm might give a better performance of the test set, but then again, a larger test set gives a more accurate representation of the performance of the algorithm.

The poor performance was due to many false positives and missed detections. The number of false positives was significantly higher for the recordings during heart surgery than for the recordings from transcatheter intervention. Consequently, a large amount of the false positives are probably due to artifacts being mistaken for air bubbles by the algorithm, as recordings of HS are more exposed to artifacts than under catheter intervention.

Although the ratio of false detections was high in this project, it was still lower than the ratio of false detections from Kjelsaas's study. However, in Kjelsaas's study, only two patients during heart surgery were included. Thus, a comparison with the results from Kjelsaas will be more representative when only including the results from the patients during catheter intervention. While the ratio of false positives was around 82% in Kjelsaas's study, it was 33.7% for the recordings of catheter intervention in this study. This suggests that the increase of a higher threshold for the detection, along with a method for handling cyclic variations, has successfully reduced the number of false positives.

The large ratio of missed detections can be explained by the difference in how the EBR is calculated by the algorithm and how it is calculated for the manual detections. For the manual detections, EBR is calculated as the highest EBR-value close to the area where Leth-Olsen et al. has marked a detection in the Color M-mode image. As seen in Figure 4.3, the position of the manually marked detections does not, in most cases, match the position of the automatic detection precisely. As observed from Figure 5.4, the detections that were detected by both the Leth-Olsen et al. and by the algorithm, there is a clear difference in the calculated EBR of the algorithm and the manual detections. When including all the detections performed manually, independent of EBR size, the number of matching detections between the algorithm and the manual detections increases, meaning that detections that Leth-Olsen et al. have estimated to have EBR below 15 dB were detected by the algorithm. Since the algorithm filters out the manual detections below 15 dB, it explains why some of the bubbles are classified as missed.

Lastly, another reason why there might be a mismatch between the detections performed by the algorithm and the manual detections is because of how the algorithm corrects the final number of detections. Because the algorithm is multi-gated, but the detection process is done one depth at a time, it has to include a step for the correction of duplicates. Bubble signals might stretch over several depths. Thus, the amplitude of the power signal might also be above the EBR threshold in several depths, causing several detections of the same

bubble. By eliminating detections that are close in depth and time, the duplicate detections are corrected. This might cause missed detections if there are multiple bubbles situated very closely. Then, the algorithm might mistake these for being the signal of the same bubble. The opposite situation might also occur, that the algorithm might end with several detections of a single bubble. This can occur when a bubble signal stretch over a longer distance in depth than what is accounted for in the algorithm. Thus, some detections might be classified as false, even though there is an actual bubble signal at the location of the detection.

6.5 RUNNING TIME

From Table 7 and Table 8, an overview of the running time of the recordings of the training set and the test set is presented. The average running time per 60 seconds was 7.6 seconds for the training set. On the other hand, the average running time per 60 seconds for the test set is only 3.3 seconds. The reason for the test set performing on a lower average running time can be explained by the number of recordings. Both the training set and the test set have around 7-8 records that have an average running time per 60 seconds of more than 10 seconds. However, for the test set, there are even more recordings with lower running time, making the outliers in running time less significant. Thus, mainly the result of the test set should be used for the evaluation of the running time of the algorithm. As seen from Table 8, there are some situations where the algorithm can use up to 1/6 of the record time. These situations often involve records with durations longer than 60 seconds, and of many artifacts, as every location with artifacts is checked when performing the detection process of the algorithm. The contrary is also true, that for some cases, such as with pulsatile blood flow the running time of the algorithm will be shortened because fewer depths must go through the detection process. Thus, one can conclude that the average running time for the algorithm is quite fast.

Part VII

CONCLUSION

7 CONCLUSION

The developed algorithm in this project detects air bubbles in the cerebral circulation based on the amplitude of the power signal of ultrasound recordings depth by depth. Any point of the power signal with an amplitude above a set EBR threshold will be detected.

The algorithm developed in this project was adjusted according to a training set of 68 recordings from two patients, where one patient was during transcatheter intervention, and the other patient was during heart surgery. The results of the training set were 75 detections in total, where 22 of the detections were from the catheter intervention, and 53 were from heart surgery. Of the 75 detections, 69.3% were correct, and 30.6% were false. The algorithm also missed 40% of the manual detections counted by Leth-Olsen et al.

For the testing of the algorithm, 582 recordings were included, where 247 of the recordings were during transcatheter intervention from 14 different patients, and 370 of the recordings were during heart surgery from 12 different patients. In total, the algorithm detected 351 bubbles in the test set, where 31.3% of the detections were correct. The ratio for false detections was 68.7%, and 70.2% of the manual detections were missed.

The results of the training set are far better than the results of the test set. The reason for this might be the ratio between the training set and the test set. Still, both the training set and the test set had a large amount of missed detections and false detections, partly due to artifacts and strict descriptions of how a signal of air emboli should be. However, the number of correct detections and false detections were improved when comparing the detections of the algorithm with all of the manual detections, and not only the manual detections with a calculated EBR above the inclusion threshold of 10.5 dB, revealing a significant mismatch between the calculated EBR of the manual detections and the EBR of the automatic detections. Consequently, the measurement of the performance of the automatic detections in comparison with the manual detections becomes less representative.

The algorithm created by Kjelsaas in her study has a detection process based on similar principles and the same recordings as in this project. However, a lower EBR threshold was used in her study than the 15 dB used in this project. With a higher EBR threshold used in this project, the results have been improved in comparison with the results of Kjelsaas' algorithm, suggesting that a focus on larger emboli gives better performance than including search of smaller bubbles as well. Furthermore, the ratio of false positives is significantly lower than

in Kjelsaas's study. Included in this project is a method for handling cyclic variations, which were Kjelsaas's primary cause of the false detections, making it possible to conclude that the method for handling cyclic variations has successfully reduced the number of false positives. However, the trade-off of the reduction in false detections has caused an increase in missed detections.

The average running time of the test set was 6%. This suggests a very fast running time and makes the algorithm eligible for use in real-time.

7.1 FUTURE WORK AND IMPROVEMENTS

For the achievement of a better performance of the algorithm developed in this project, some possibilities include training the algorithm on even more combinations of the values of the adjustable parameters in the algorithm. Furthermore, better performance of the test set might be obtained by training the algorithm on a larger size of the training set, making the algorithm more familiar with even more various cases of ultrasound recordings.

The ratio of false detections might also be reduced by the implementation of machine learning in the form of image recognition. This proposal is inspired from Guepie et al.'s study on "Sequential emboli detection from ultrasound outpatient data," which is previously mentioned in the literature review, which combines the use of a single-gated detection algorithm with the use of a classification algorithm. The detection detects all positions of the signal where the intensity is higher than a chosen threshold. Then, the classification algorithm handles the discrimination of artifacts and air bubbles through the use of machine learning with the recognition of features from the Color M-Mode and the Doppler spectrogram. The same idea implemented with machine learning of classification of artifacts and air bubbles in the Color M-Mode image be able to improve the results when combined with the algorithm for detection of air emboli.

BIBLIOGRAPHY

[1] Bjørn Atle Johan Angelsen and Hans Garman Torp. *Ultrasound Imaging: Waves, Signals, and Signal Processing*. Vol. 1. 2000. URL: https://books.google.no/books/about/Ultrasound_Imaging.html?id=celDXwAACAAJ&redir_esc=y.

[2] *Anterior Cerebral Artery Anatomy, Function & Diagram*. URL: https://www.healthline.com/human-body-maps/anterior-cerebral-artery.

[3] Henny H. Billett. "Hemoglobin and Hematocrit." In: *Anesthesiology* 28.4 (July 1967), pp. 763–763. ISSN: 0003-3022. DOI: 10.1097/00000542-196707000-00028. URL: https://www.ncbi.nlm.nih.gov/books/NBK259/.

[4] Emma M.L. Chung, Caroline Banahan, Nikil Patel, Justyna Janus, David Marshall, Mark A. Horsfield, Clément Rousseau, Jonathan Keelan, David H. Evans, and James P. Hague. *Size Distribution of Air Bubbles Entering the Brain During Cardiac Surgery*. Tech. rep. 4. Apr. 2015. DOI: 10.1371/journal.pone.0122166.

[5] Richard S. C. Cobbold. *Foundations of Biomedical Ultrasound*. Oxford University Press, 2006. ISBN: 9780195168310. URL: https://global.oup.com/academic/product/foundations-of-biomedical-ultrasound-9780195168310?cc=no&lang=en&.

[6] Marisa Cullinane et al. "Evaluation of New Online Automated Embolic Signal Detection Algorithm, Including Comparison With Panel of International Experts." In: *Stroke* 31.6 (2000), pp. 1335–1341. DOI: 10.1161/01.STR.31.6.1335. URL: https://www.ahajournals.org/doi/abs/10.1161/01.STR.31.6.1335.

[7] Stephanie Gordy and Susan Rowell. "Vascular air embolism." In: *International Journal of Critical Illness and Injury Science* 3.1 (2013), p. 73. ISSN: 2229-5151. DOI: 10.4103/2229-5151.109428. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3665124/.

[8] Blaise Kevin Guepie, Matthieu Martin, Victor Lacrosaz, Marilys Almar, Benoit Guibert, and Philippe Delachartre. "Sequential emboli detection from ultrasound outpatient data." In: *IEEE Journal of Biomedical and Health Informatics* 23.1 (Jan. 2019), pp. 334–341. DOI: 10.1109/JBHI.2018.2808413.

[9] Lars Hoff. *Acoustic Characterization of Contrast Agents for Medical Ultrasound Imaging*. Tech. rep. 2000.

[10] N. F. Kirkby. "Polytropic Process." In: *A-to-Z Guide to Thermodynamics, Heat and Mass Transfer, and Fluids Engineering*. Begellhouse, June 2011. DOI: 10.1615/atoz.p.polytropic{_}process. URL: https://www.thermopedia.com/content/1045/.

[11] Karoline Kjelsaas. *Detection of Air Emboli in the Brain of Neonates by Ultrasound Doppler*. Tech. rep. Faculty of Information Technology and Electrical Engineering, Department of Electronic Systems, NTNU, Aug. 2020.

[12] My Tam Lam. *Estimation of Air Bubbles in The Cerebral Circulation in Newborns By Ultrasound Doppler Technique*. Tech. rep. Trondheim: Department of Physics, Faculty of Natural Sciences, NTNU, 2021.

[13] Martin Leth-Olsen, Gaute Døhlen, Hans Torp, and Ann Nyrnes. *Detection of Cerebral High Intensity Transient Signals by NeoDoppler During Cardiac Catheterization and Cardiac Surgery in Infants*. Tech. rep. 2021.

[14] H Markus, A Loh, and M M Brown. "Computerized Detection of Cerebral Emboli and Discrimination from Artifact Using Doppler ultrasound." In: *Stroke* 24.11 (1993), pp. 1667–1672. DOI: 10.1161/01.STR.24.11.1667. URL: https://www.ahajournals.org/doi/abs/10.1161/01.STR.24.11.1667.

[15] Matthew J. Martin, Emma M.L. Chung, Kumar V. Ramnarine, Alison H. Goodall, A. Ross Naylor, and David H. Evans. "Thrombus size and doppler embolic signal intensity." In: *Cerebrovascular Diseases* 28.4 (Sept. 2009), pp. 397–405. ISSN: 10159770. DOI: 10.1159/000235627.

[16] Colin J. McCarthy, Sasan Behravesh, Sailendra G. Naidu, and Rahmi Oklu. "Air Embolism: Practical Tips for Prevention and Treatment." In: *Journal of Clinical Medicine* 5.11 (Nov. 2016). ISSN: 20770383. DOI: 10.3390/JCM5110093. URL: /pmc/articles/PMC5126790//pmc/articles/PMC5126790/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC5126790/.

[17] Mark A Moehring and John R Klepper. *Pulse Doppler Ultrasound Detection, Characterization and Size Estimation of Emboli in Flowing Blood*. Tech. rep. 1994.

[18] R. S. Moorthy. "Doppler ultrasound." In: *Medical Journal Armed Forces India* 58.1 (2002), pp. 1–2. ISSN: 03771237. DOI: 10.1016/S0377-1237(02)80001-6. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4923974/.

[19] Andrew Murphy and Matt Morgan. "Acoustic impedance." In: *Radiopaedia.org* (Nov. 2014). ISSN: 0044-4650. DOI: 10.53347/RID-32118. URL: https://radiopaedia.org/articles/acoustic-impedance.

[20] B. P. Murphy, F. J. Harford, and F. S. Cramer. "Cerebral air embolism resulting from invasive medical procedures." In: *Annals of Surgery* 201.2 (1985), pp. 242–245. ISSN: 00034932. DOI: 10.1097/00000658-198502000-00019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1250649/.

[21] John William Strutt Rayleigh. *The Theory of Sound*. London: Macmillian and co., 1877.

[22] Asbjorn Stoylen. *Basic ultrasound, echocardiography and Doppler ultrasound*. Accessed: 19.05.2021. URL: https://folk.ntnu.no/stoylen/strainrate/Basic_Doppler_ultrasound.

[23] Tak W. Mak, Mary E. Saunders, and Bradley D. Jett. "Transplantation." In: *Primer to the Immune Response*. Second Edition. Elsevier, Jan. 2014. Chap. 17, pp. 457–486. DOI: 10.1016/B978-0-12-385245-8.00017-0. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780123852458000170.

Part VIII

APPENDIX

APPENDIX

This chapter includes the Matlab scripts used in this project.

A.1 THE ALGORITHM FOR THE DETECTION FOR AIR EMBOLI

This section includes the complete algorithm for the detection of air emboli in the cerebral circulation developed in this project. The algorithm is displayed in Listing 6.

Listing 6: The Algorithm For The Detection of Air Emboli

```
% BubbleInspect

%% Set input variables
% cut-off frequency
fc = 20.3; %20.3
% n0 = Start depth for detection
n0 = 8;  % 8 % 7
% N = number of points to calculate the movmean of background
    ↪ signal
N = 500; %250; 500; 1000; 2500, 4069,
% N_art = number of points to calculate the movmean of artefact
    ↪ treshold
N_art = 500; %250, 500, 1000, 2500, 5000, 10000
% percentage of backgroundsignal to use for artefact calculation
threshArt = 0.3; % 0.5 %0.25 % p r v 0.25, 0.40, 0.5, 0.6, 0.75
% minArt_tDist = minimum time distance for it to be an artefact
artWidth = 0.1; % 0.2 %0.10 %0.05 p r ve fra 0.01, 0.05, 0.1,
    ↪ 0.15, 0.2

tMinD = 0.2;     %0.2 % seconds
tMaxD = 0.60;    %0.6 % seconds
tMinN = 0.075;   %0.075% seconds
zMin  = 5;       %5 % mm
zMax  = 11.5;    %11.5   % mm

stdLim = 4; % 3, 3.5, 4
% EBRmin = set threshold value
EBRmin = 15; % 15, 17.5, 20, 22.5, 20

%% Load folder with recordings
PathName = uigetdir(cd);
if isequal(PathName,0), return; end
cd(PathName);
fInfo = dir('20*.mat');
NRec  = length(fInfo);
```

```
% Remove last path-folder for easy patient number access
if (strfind(PathName,'Kirurgi')~=0)
    PathName = PathName(1:(strfind(PathName,'Kirurgi')-2));
elseif (strfind(PathName,'Pre')~=0)
    PathName = PathName(1:(strfind(PathName,'Pre')-2));
elseif (strfind(PathName,'Post')~=0)
    PathName = PathName(1:(strfind( PathName,'Post')-2));
end

%% Begin looping through record by record
Detected = 0;
Manual   = 0;
Correct  = 0;
Missed   = 0;
False    = 0;

wholeLoop = tic;
for nRec = (1):NRec
    oneLoopTime = tic;
    disp(newline);
    fileName = fInfo(1).name;
    disp(fileName);
    load(fileName,'Cmmode','p');

    % Struct with recording info
    rec = struct('PathName',PathName,'fileName',fileName,'cmmode'
        ↪ ,Cmmode,'p',p);

    %% ------------------------ Get signal details
        ↪ ----------------------
    % Struct with signal info
    sig = struct();
    % Pow in dB for all depths
    sig.pow_dB  = rec.cmmode.dBStep*double(rec.cmmode.PdB);
    [nd,nt] = size(sig.pow_dB);              % Depth, timepoints
    sig.nd = nd;
    depthInd= 1:nd;                                % Get depths
    sig.z       = rec.cmmode.depthAx*1000;   % Get depth axis in
        ↪ [mm]
    sig.t       = double(rec.cmmode.timeAx);         % Get time
        ↪ axis in [s]
    sig.zIncr   = sig.z(2)-sig.z(1); % Get depth incr in [mm]
    sig.tIncr   = sig.t(2)-sig.t(1);   % Get time incr in [s]

    %% ------------------------ Low pass filter
        ↪ --------------------------
    prf    = 1/sig.tIncr;
    %fc     = 20.3;
    [b,a]  = butter(2,fc/(prf/2));     % Butterworth filter of 2
        ↪ nd order
    %sig.pow_dB = filter(b,a,sig.pow_dB,[],2);
    %figm   = figure(26);
```

```
%imagesc(sig.t,sig.z,sig.pow_dB); colormap gray;
%title(fileName(1:15) + ": M-Mode"); % of Filtered signal");
%xlabel("Time [s]"); ylabel("Depth [mm]");

%% ---------------------- Calculate Doppler Shift
    ↪ -----------------------
sig.vel = double(rec.cmmode.fi)/128*double(rec.cmmode.
    ↪ vNyquist); %max value +/-128
sig.cb = 1570; sig.f0 = 7812500; %From Eb
sig.fd = (2*sig.f0*sig.vel)/sig.cb; sig.T = abs(10./sig.fd);
%%
%[bgs_n] = estimateBgs(rec,sig,sig.pow_dB(22,:),22,N,stdLim);
    ↪  %19,7

%% --------------------------- Find artifacts
    ↪ ---------------------------

[art, sig] = findArtLoc(sig,rec,threshArt,N_art);

%% --------------------- Detect embolic signals
    ↪ ------------------------
%detecStart = tic;                  % 500 = 1.22 sec, to
    ↪ hjertebank

res = detectEmb(sig,rec, art, EBRmin, n0, N, artWidth, stdLim
    ↪ );

%% ----------------------- Get manually detected
    ↪ -----------------------
manu = getManual(rec, sig, res, EBRmin, n0);

%% ------- Draw detected embolic signals VS manually detected
    ↪  ---------

%drawDetections(rec,sig,EBRmin,auto0,manu,"Before");

% ---------------------- Check for duplicates
    ↪ -------------------------
%tic;
res = correctDuplicates(sig,res,tMinD,tMaxD,tMinN,zMin,zMax);
%toc;

% ------------- Draw detected embolic signals in cmmode
    ↪ --------------
[auto,TableBub]=changeFormat(sig,rec,res,fc,n0,N,N_art,
    ↪ threshArt,artWidth,stdLim,EBRmin);
drawDetections(rec, sig, EBRmin, auto, manu, "After");

%% ------------------------- Compare results
    ↪ --------------------------
comp = compare(res,auto,manu,tMinD,tMaxD,zMin,zMax);
%comp = newCompare(res,auto,manu,tMinD,tMaxD,zMin,zMax);
```

```
%toc;

%% ------------------------- Save 2 excel ----------------------------
%saveRec2excel(rec, auto, TableBub);
%saveManual2Excel(rec,comp,EBRmin);


%% Save this recordings' counts
De = res.bubCount_all;
Ma = manu.manCount;
Co  = length(comp.tCorrect);
Mi  = length(comp.tMissed);
Fa = length(comp.tFalse);
pasNum = (str2num(PathName((end-1):end))); % vurdere    bytte til str2double for raskere ytelse
tRec = rec.cmmode.timeAx(end);
if (strfind(PathName,'Kat')~=0)
    type = 1;%'Kat';
else

    type = 2;%'Kir';
end
fileLocStr='/Users/mytamlam/Dropbox/2021_MyLamProsjekt/Kode';
fname = fullfile(fileLocStr,'OverviewDetections_test_w_all_manual.xlsx');
tOneLoop = toc(oneLoopTime);
disp("Time: "+tOneLoop);
if (isfile(fname))
    TableP = readtable(fname,'Sheet',2);
    newRow = {type,pasNum,{rec.fileName},fc,n0,N,N_art,threshArt,artWidth,tMinD,tMaxD,tMinN,zMin,zMax,stdLim,EBRmin,De,Ma,Co,Mi,Fa,tRec,tOneLoop};
    newTableP = [TableP;newRow];
    newTableP.Properties.VariableNames={'type','pasNum','fileName','fc','n0','N','N_art','threshArt','artWidth','tMinDiag','tMaxDiag','tMinNorm','zMin','zMax','stdLim','EBRmin','Detected','Manual','Correct','Missed','False','tRec','runT'};
    writetable(newTableP, fname, 'Sheet', 2);
    disp('Updated table with this rec.');
else
    TableP = table(type,pasNum,{rec.fileName},fc,n0,N,N_art,threshArt,artWidth,tMinD,tMaxD,tMinN,zMin,zMax,stdLim,EBRmin,De,Ma,Co,Mi,Fa,tRec,tOneLoop);
    TableP.Properties.VariableNames={'type','pasNum','fileName','fc','n0','N','N_art','threshArt','artWidth','tMinDiag','tMaxDiag','tMinNorm','zMin','zMax','stdLim','EBRmin','Detected','Manual','Correct','Missed','False','tRec','runT'};
    writetable(TableP, fname, 'Sheet', 2);
    disp('Created table.');
```

```
    end


    %% ----------------------- Update total count
        ↪ -------------------------
    Detected = Detected + res.bubCount_all;
    Manual   = Manual + manu.manCount;
    Correct  = Correct + length(comp.tCorrect);
    Missed   = Missed + length(comp.tMissed);
    False    = False + length(comp.tFalse);
    %toc;
end
endWholeLoop = toc(wholeLoop);
disp("Total time: "+endWholeLoop);

%% Save overview of recordings of Pasient X to excel
% vurdere    bytte til str2double for raskere ytelse
pasNum = (str2num(PathName((end-1):end))); % vurdere    bytte til
    ↪  str2double for raskere ytelse
if (strfind(PathName,'Kat')~=0)
    type = 1;%'Kat';
else
    type = 2;%'Kir';
end
fileLocStr='/Users/mytamlam/Dropbox/2021_MyLamProsjekt/Kode';
fname = fullfile(fileLocStr,'OverviewDetections_test_w_all_manual
    ↪ .xlsx');
if (isfile(fname))
    TableP = readtable(fname,'Sheet',1);
    newRow = {type,pasNum,fc,n0,N,N_art,threshArt,artWidth,tMinD,
        ↪ tMaxD,tMinN,zMin,zMax,stdLim,EBRmin,Detected,Manual,
        ↪ Correct,Missed,False};
    newTableP = [TableP;newRow];
    newTableP.Properties.VariableNames={'type','pasNum','fc','n0'
        ↪ ,'N','N_art','threshArt','artWidth','tMinDiag','
        ↪ tMaxDiag','tMinNorm','zMin','zMax','stdLim','EBRmin','
        ↪ Detected','Manual','Correct','Missed','False'};
    writetable(newTableP, fname, 'Sheet', 1);
    disp('Updated table.');
    disp('-----------------------------------------');
else
    TableP = table(type,pasNum,fc,n0,N,N_art,threshArt,artWidth,
        ↪ tMinD,tMaxD,tMinN,zMin,zMax,stdLim,EBRmin,Detected,
        ↪ Manual,Correct,Missed,False);
    TableP.Properties.VariableNames={'type','pasNum','fc','n0','N
        ↪ ','N_art','threshArt','artWidth','tMinDiag','tMaxDiag'
        ↪ ,'tMinNorm','zMin','zMax','stdLim','EBRmin','Detected'
        ↪ ,'Manual','Correct','Missed','False'};
    writetable(TableP, fname, 'Sheet', 1);
    disp('Created table.');
    disp('-----------------------------------------');
end
```

```
load handel;
%beep;
sound(y,Fs);

%% FUNCTIONS

% FIND ARTIFACT LOCATIONS
function [art, sig] = findArtLoc(sig,rec,threshArt,N_art)
    fileName = rec.fileName;
    pow_dB   = sig.pow_dB;
    t        = sig.t;

    meanPow  = mean(pow_dB,1);  %(1:7,:),1);
    meanmean = 4;%mean(meanPow)*(1-(thresh_val)+.10);
    artLoc1  = meanPow>(movmedian(meanPow,N_art)*(1+threshArt));
    artLoc2  = meanPow < meanmean; %movmeanBgs < meanmean;
    artLoc   = zeros(1,length(meanPow));

    for i = 1:length(meanPow)
       if (artLoc1(i)==1) || (artLoc2(i)==1)
           artLoc(i) = 1;
       end
    end

    sig.startInd = 1;
    while (artLoc(sig.startInd)==1)
        if (sig.startInd == length(artLoc))
            break
        end
        sig.startInd = sig.startInd + 1;
    end

    sig.endInd = length(artLoc);
    while (artLoc(sig.endInd)==1)
        if (sig.endInd == 1)
            break
        end
        sig.endInd = sig.endInd - 1;
    end

    artTime0     = artLoc.*t;
    art          = struct();
    art.artTime  = artTime0(artTime0~=0);
    art.artCount = length(art.artTime);

    %figgi = figure(1); %str2double(fileName(12:15)));
    %plot(t,meanPow,'color', [.59, .82, .83]); hold on;
    %yline(meanmean,'--','color',[.96 .67 .40]); hold on;
    %plot(t,movmedian(meanPow,N_art)*(1+threshArt)); hold off;
    %title(fileName(1:15) + ": Mean power of all depths");
    %legend("Power signal", "Meanmean", "Artifact treshold");% "
        ↪ Meanmean");%, "Artifact location");
```

```
end

% ESTIMATE BACKGROUND SIGNALS
function [bgs_n] = estimateBgs(rec, sig, pow_n, n, N, stdLim)
    t = sig.t; zIncr = sig.zIncr; fileName = rec.fileName;

    % Exclude inadequate depths from being used in detection
    %nbins = 5;
    %[counts, centers] = hist(pow_n,nbins);
    %figure();
    %bar(centers,counts,'FaceColor','[.18, .46, .47]');
    %title(fileName(1:15)+" Histogram depth "+n*zIncr);
    %xlabel('Amplitude [dB]');

    %highSpread = (sum(counts>(sum(counts)*0.05))) >= (nbins/2);
    STD = std(pow_n);
    disp(STD);
    highSpread = STD > stdLim;

    % Check if highSpread, then exclude fromd estimation
    if highSpread
        bgs_n = NaN;
        %disp(bgs_n);
    else
        % Exclude start and end points from being used in
            ↪ estimation of bgs
        pow_fix = pow_n;
        iStart  = max([sig.startInd,30]); % 30 timepoints * tIncr
            ↪  = 0.1 s
        iEnd    = min([sig.endInd,(length(t)-30)]);

        for i = 1:length(pow_n)
            if (i<iStart) || (i>iEnd)
                pow_fix(i) = NaN;
            end
        end
        pow_fix = fillmissing(pow_fix,'movmean',N);

        % Exclude peaks from estimation of bgs
        bgs_pos = pow_fix < (movmedian(pow_fix,N)*1.4); %(
            ↪ medfilt1(pow_n, N)*1.4);
        hollow_bgs = pow_fix;

        for i = 1:(length(pow_fix)-1)
            if bgs_pos(i)==0
                hollow_bgs(i)=NaN;
            end
        end
        % fill the removed parts by neighbouring means
        filled_bgs = fillmissing(hollow_bgs, 'movmean', N);
        % calculate the moving mean of the background signal
        bgs_n = movmean(filled_bgs, N);
```

```
        %figure();
        %plot(t,pow_n,'color', [.59, .82, .83]); hold on;
        %%plot(t,pow_fix); hold on;
        %%plot(t,movmedian(pow_fix,N)); hold on;
        %plot(t,movmedian(pow_fix,N)*1.4); hold on;
        %plot(t,bgs_n,'color', [.18, .46, .47]); hold on;
        %plot (t,bgs_n+15,'color',[.96 .67 .40]);hold off;
        %title(fileName(1:15)+" Calculation of bgs signal in
            ↪ depth "+n*zIncr);
        %%legend("Power signal","pow fix","Movmedian", "Movmedian
            ↪  * 1.4", "Bgs", "Bgs+15");% "medfilt1", "Meanmean
            ↪ ");%, "Artifact location");
        %legend("Power signal","Bgs","bgs+15");
    end
end

% DETECT EMBOLIC SIGNALS
function [res] = detectEmb(sig,rec,art,EBRmin,n0,N,artWidth,
    ↪ stdLim)
    pow_dB = sig.pow_dB; nd = sig.nd;
    tIncr = sig.tIncr; zIncr = sig.zIncr;
    t = sig.t; T = sig.T;
    startInd = sig.startInd; endInd = sig.endInd;
    artTime = art.artTime;
    %minBubLength = 2;
    %artWidth = 0.05;

    % Create struct to save results
    res = struct();

    bubSig_all = {};   bgs_all = {};
    thresh_all = {};   embCount_all = 0;
    timeIdx_all = {};  bubAmp_all = {};
    Ibub_all = {};

    iStart = max([startInd,30]); % 30 timepoints * tIncr = 0.1 s
    iEnd   = min([endInd,(length(t)-30)]);
    for n = n0:nd
        %disp("n " + n);
        pow_n = pow_dB(n,:);
        [bgs_n] = estimateBgs(rec, sig, pow_n, n, N, stdLim);
        bgs_all{n} = bgs_n;
        thresh = bgs_n + EBRmin;
        thresh_all{n} = thresh;
        bub_sig_n = NaN([1 length(pow_n)]);
        bub = 0;
        emb_count = 0;          % Count of embolic signals for
            ↪ depth n
        check_bgs = (~isnan(bgs_n));

        if (any(check_bgs))      % Skip depth if pulsative signal
```

```
        j = 1;
        for i=iStart:iEnd                % Avoid searching in
            ↪ the far ends of the signal
            % find closest value in list of artifact time
                ↪ positions:
            ti = i*tIncr;
            [minVal, closestIndex] = min(abs(artTime-ti));
            closestArtTime = artTime(closestIndex);
            expectedLength = round(T(n,i)/tIncr);
            bubLength = 0;
            if (abs(closestArtTime - ti) > artWidth)% Ensure
                ↪ not close to artifact
                if (i >= j && pow_n(i)>thresh(i) && j<=iEnd)
                    j = i;
                    bubLength = 1;
                    tj = ti;
                    while (pow_n(j) > thresh(j)) && (abs(
                        ↪ closestArtTime-tj)>artWidth) && (j<
                        ↪ iEnd)
                        bubLength = bubLength + 1;
                        j = j + 1;
                        tj = j*tIncr;
                        [minVal, closestIndex] = min(abs(
                            ↪ artTime-tj));
                        closestArtTime = artTime(closestIndex)
                            ↪ ;
                    end
                    if bubLength > expectedLength %/
                        ↪ minBubLength
                        bub = bub + 1;
                        [val, idx] = max(pow_n(i:j));   % get
                            ↪ peak index and value
                        idx = idx + (i-1);
                        bub_sig_n(idx) = val;           % save
                            ↪  power of bub peak to correct
                            ↪ index
                        emb_count = emb_count + 1;       % add
                            ↪  1 bubble to count
                    end
                end
            end
        end
    end

    embCount_all = embCount_all + emb_count;
    bubSig_all{n} = bub_sig_n;

    figure(); %ax(1)=subplot(2,1,1);
    plot(t, pow_n, 'color', [.59, .82, .83]); hold on; %
        ↪ lysturkis
    plot(t, bgs_n, 'color', [.18, .46, .47]); hold on; %
        ↪ morkturkis
```

```
        plot(t, thresh, 'color', [.84,.87,.49]); hold on; %gr nn
        plot(t, bub_sig_n, 'oy','MarkerSize',12,'MarkerEdgeColor'
            ↪ ,[.96,.67,.40]); %[0.27,0.73,0.75]);
        hold off;
        title("Amplitude of signal in depth " + n*zIncr + " mm");
        xlabel("Time [s]"); ylabel("Power [dB]");
        lgd = legend("Power signal", "Filtered bgsignal w/
            ↪ movmean of " + N + " points", "Threshold", "
            ↪ Detected");
        %lgd = legend("Power signal", "Filtered background signal
            ↪ ", "Threshold", "Detected");
        lgd.Location = 'southeast';

        % Print text with bubbles
        % find() gives index-number
        Ibub = find(~isnan(bubSig_all{n}));
        Ibub_all{n} = Ibub;
        timeIdx = Ibub * tIncr;
        timeIdx_all{n} = timeIdx;
        bubAmp = bub_sig_n(~isnan(bub_sig_n));
        bubAmp_all{n} = bubAmp;

    end
    res.bubSig_all = bubSig_all;
    res.bgs_all = bgs_all;
    res.thresh_all = thresh_all;
    res.embCount_all = embCount_all;
    res.timeIdx_all = timeIdx_all;
    res.bubAmp_all = bubAmp_all;
    res.Ibub_all = Ibub_all;
end

% CORRECT DUPLICATES
function res = correctDuplicates(sig,res,tMinD,tMaxD,tMinN,zMin,
    ↪ zMax)
    timeIdx_all  = res.timeIdx_all;
    bubAmp_all   = res.bubAmp_all;
    bubCount_all = res.embCount_all;

    nd = sig.nd; zIncr = sig.zIncr; tIncr = sig.tIncr;

    for n1 = 1:nd
        tBub1 = timeIdx_all{n1};
        for n2 = 1:nd
            tBub2 = timeIdx_all{n2};
            for i = 1:length(tBub1)
                for j = 1:length(tBub2)
                    tDiff = abs(tBub1(i) - tBub2(j)); %  && (n1~=
                        ↪ n2)
                    zDiff = abs(n1*zIncr-n2*zIncr);
                    bub1exists = ~isnan(timeIdx_all{n1}(i));
                    bub2exists = ~isnan(timeIdx_all{n2}(j));
```

```
                % remove duplicates of same bubble in close
                    ↪ diagonal direction
                if (n1~=n2) && (zDiff<zMin) && (tDiff<tMinD)
                    ↪ && (bub2exists) && (bub1exists)
                    if (bubAmp_all{n1}(i) < bubAmp_all{n2}(j)
                        ↪ )
                        timeIdx_all{n1}(i) = NaN; bubAmp_all{
                            ↪ n1}(i) = NaN;
                        bubCount_all = bubCount_all - 1;
                    else
                        timeIdx_all{n2}(j) = NaN; bubAmp_all{
                            ↪ n2}(j) = NaN;
                        bubCount_all = bubCount_all - 1;
                    end

                % remove duplicates of same bubbles in
                    ↪ further diagonal
                % distance
                elseif (n1~=n2) && (zDiff>=zMin) && (zDiff<
                    ↪ zMax) && (tDiff>tMinD) && (tDiff<tMaxD
                    ↪ ) && (bub2exists) && (bub1exists)
                    if (bubAmp_all{n1}(i) < bubAmp_all{n2}(j)
                        ↪ )
                        timeIdx_all{n1}(i) = NaN; bubAmp_all{
                            ↪ n1}(i) = NaN;
                        bubCount_all = bubCount_all - 1;
                    else
                        timeIdx_all{n2}(j) = NaN; bubAmp_all{
                            ↪ n2}(j) = NaN;
                        bubCount_all = bubCount_all - 1;
                    end

                % remove duplicates of same bubble in
                    ↪ horizontal direction
                elseif (n1==n2) && (i~=j) && (tDiff<tMinN) &&
                    ↪  (bub2exists) && (bub1exists)
                    if (bubAmp_all{n1}(i) < bubAmp_all{n2}(j)
                        ↪ )
                        timeIdx_all{n1}(i) = NaN; bubAmp_all{
                            ↪ n1}(i) = NaN;
                        bubCount_all = bubCount_all - 1;
                    else
                        timeIdx_all{n2}(j) = NaN; bubAmp_all{
                            ↪ n2}(j) = NaN;
                        bubCount_all = bubCount_all - 1;
                    end
                end
            end
        end
    end
end
```

```
    res.timeIdx_all = timeIdx_all;
    res.bubAmp_all = bubAmp_all;
    res.bubCount_all = bubCount_all;
    res.tMinD = tMinD;
    res.tMaxD = tMaxD;
    res.tMinN = tMinN;
    res.zMin  = zMin;
    res.zMax  = zMax;
end

% CONVERT TO EXCEL FORMAT
function [auto,TableBub]=changeFormat(sig,rec,res,fc,n0,N,N_art,
    ↪ threshArt,artWidth,stdLim,EBRmin)
    bubAmp_all = res.bubAmp_all;    bgs_all  = res.bgs_all;
    PathName   = rec.PathName;      Ibub_all = res.Ibub_all;
    tIncr = sig.tIncr;
    zIncr = sig.zIncr;
    nd    = sig.nd;

    if (strfind(PathName,'Kat')~=0)
        type = 1;%'Kat';
    else
        type = 2;%'Kir';
    end

    tb = [];
    zb = [];
    AmpdB = [];
    %Amplitude = [];
    Background_dB = [];
    Ratio_dB = [];

    for n = 1:nd
        for i = 1:length(bubAmp_all{n})
            if (~isnan(bubAmp_all{n}(i)))
                ind = Ibub_all{n}(i);
                zb(end+1) = n * zIncr;
                tb(end+1) = ind * tIncr;
                AmpdB(end+1) = bubAmp_all{n}(i);
                %Amplitude(end+1) = 10^(AmpdB(end)/10);
                Background_dB(end+1) = bgs_all{n}(ind);
                Ratio_dB(end+1) = AmpdB(end)-Background_dB(end);
            end
        end
    end

    auto = struct();

    tb = transpose(tb);
    tNum = Sec2dateNum(tb,rec);
    zb = transpose(zb);
    AmpdB = transpose(AmpdB);
```

```
    %Amplitude = transpose(Amplitude);
    Background_dB = transpose(Background_dB);
    Ratio_dB = transpose(Ratio_dB);
    SingleBubbles = (string(datestr(tNum,'YYYY:mm:dd:HH:MM:SS:FFF
        ↪ ')));
    type = type .* ones(length(tb),1);
    pasNum = (str2num(rec.PathName((end-1):end)));
    pasNum = pasNum .* ones(length(tb),1);
    fileName = repmat({rec.fileName},length(tb),1);%.*ones(length
        ↪ (tb),1);
    fc = fc .* ones(length(tb),1);
    n0 = n0 .* ones(length(tb),1);
    N = N .* ones(length(tb),1);
    N_art = N_art * ones(length(tb),1);
    threshArt = threshArt .* ones(length(tb),1);
    artWidth = artWidth .* ones(length(tb),1);
    tMinD = res.tMinD .* ones(length(tb),1);
    tMaxD = res.tMaxD .* ones(length(tb),1);
    tMinN = res.tMinN .* ones(length(tb),1);
    zMin = res.zMin .* ones(length(tb),1);
    zMax = res.zMax .* ones(length(tb),1);
    stdLim = stdLim .* ones(length(tb),1);
    EBRmin = EBRmin .* ones(length(tb),1);


    TableBub = table(type,pasNum,fileName,fc,n0,N,N_art,threshArt
        ↪ ,artWidth,tMinD,tMaxD,tMinN,zMin,zMax,stdLim,EBRmin,
        ↪ SingleBubbles,tNum,zb,AmpdB,Background_dB,Ratio_dB);

    auto.tb = tb; auto.zb = zb;
    auto.AmpdB = AmpdB; auto.Ratio_dB = Ratio_dB;
end

% GET INFO ON MANUAL DETECTIONS
function manu = getManual(rec, sig, res, EBRmin, n0)
    PathName = rec.PathName; fileName = rec.fileName;
    tIncr = sig.tIncr; zIncr = sig.zIncr;
    pow_dB = sig.pow_dB;
    % vurdere    bytte til str2double for raskere ytelse
    pasNum = int2str(str2num(PathName((end-1):end))); % vurdere
        ↪    bytte til str2double for raskere ytelse
    if (strfind(PathName,'Kat')~=0)
        type = 'Kat4e_1/Pas';
    else
        type = 'Kir4e_2/Pas';
    end
    fileLocStr=['/Users/mytamlam/Dropbox/BubbleCountwRatio/',type
        ↪ , pasNum];
    mFileName = fullfile(fileLocStr,[fileName(1:16),'Hits.xlsx'])
        ↪ ;
    tb_M = []; zb_M = []; AmpdB_M = []; Ratio_dB = [];
    if (isfile(mFileName))
```

```
        myTableBub = readtable(mFileName,'Sheet',1);
        if ~isempty(myTableBub)
            if isfield(table2struct(myTableBub),'SingleBubles')
                include = myTableBub.Ratio_dB > (EBRmin-EBRmin
                    ↪ *0.3);
                tb_M1   = myTableBub.tb(include==1);
                zb_M    = myTableBub.zb(include==1);
                tb_M    = dateNum2sec(tb_M1, rec);
                AmpdB_M = myTableBub.AmpdB(include==1);
                Ratio_dB = myTableBub.Ratio_dB(include==1);
            end
        end
    end
    manu = struct('tb',tb_M,'zb',zb_M,'AmpdB',AmpdB_M, 'manCount'
        ↪ ,length(tb_M));
    manu.Ratio_dB = Ratio_dB;
    manu.fileName = mFileName;
end


% DRAW DETECTIONS VS MANUAL DETECTIONS
function drawDetections(rec,sig,EBRmin,auto, manu,state)
    fileName = rec.fileName;
    pow_dB = sig.pow_dB;
    t = sig.t; z = sig.z;
    tb = auto.tb;   zb = auto.zb;
    tb_M = manu.tb; zb_M = manu.zb;
    figNum = str2num(fileName(10:15));

    if (~isempty(tb_M)) && (~isempty(tb)) % both exist
        figm = figure(figNum);
        imagesc(t,z,pow_dB); colormap gray; hold on;
        p1 = plot(tb,transpose(zb),'o','MarkerSize',12,'LineWidth
            ↪ ', 1.5);
        p1.Color = '[.27,.73,.75]'; hold on;
        p2 = plot(tb_M,zb_M,'* ','MarkerSize',12,'LineWidth',1.5)
            ↪ ;
        p2.Color = '[.74,.82,.15]'; hold off;
        title(fileName(1:16)+" M-Mode, EBRmin = "+EBRmin+", "+
            ↪ state);
        xlabel("Time [s]"); ylabel("Depth [mm]");
        legend("Detected bubbles", "Manually detected bubbles");
        %saveas(figm, (fileName(1:16) + ' M-Mode EBRmin ' +
            ↪ EBRmin));

    elseif (~isempty(tb_M))                % only manual detec
        figm = figure(figNum);
        imagesc(t,z,pow_dB); colormap gray; hold on;
        p2 = plot(tb_M,zb_M,'*','MarkerSize',12,'LineWidth',1.5);
        p2.Color = '[.74,.82,.15]'; hold off;
        title(fileName(1:16)+" M-Mode, EBRmin = "+EBRmin+", "+
            ↪ state);
```

```
        xlabel("Time [s]"); ylabel("Depth [mm]");
        legend("Manually detected bubbles");
        %saveas(figm, (fileName(1:16) + ' M-Mode EBRmin ' +
            ↪ EBRmin));

    elseif (~isempty(tb))                   % only auto detec

        figm = figure(figNum);
        imagesc(t,z,pow_dB); colormap gray; hold on;
        p1 = plot(tb,transpose(zb),'o','MarkerSize',12,'LineWidth
            ↪ ',1.5);
        p1.Color = '[.27,.73,.75]'; hold off;
        title(fileName(1:16)+" M-Mode, EBRmin = "+EBRmin+", "+
            ↪ state);
        xlabel("Time [s]"); ylabel("Depth [mm]");
        legend("Detected bubbles");
        %saveas(figm, (fileName(1:16) + ' M-Mode EBRmin ' +
            ↪ EBRmin));

        %disp('There were zero manually counted bubbles.');

    else
        disp('There were zero bubbles detected, both by the
            ↪ algorithm and manually.');
    end
end

% DRAW ONLY DETECTIONS
function figm = drawOnlyDetections(figNum,state,tb,zb)
    global fileName; global EBRmin; global t; global z; global
        ↪ pow_dB;
    %ob = guidata(figh);
    figm = figure(figNum); %ax(1) = subplot(2,1,1);        % Plot
        ↪  m-mode
    imagesc(t,z,pow_dB); colormap gray; hold on;
    plot(tb,transpose(zb),'o','MarkerSize',12,'MarkerEdgeColor',
        ↪ [.27, .73, .75], 'LineWidth', 1.5); hold off; %, '
        ↪ MarkerFaceColor', 'r'
    title(fileName(1:16) + " M-Mode, " + "EBRmin = " + EBRmin +
        ↪ ", " + state); xlabel("Time [s]"); ylabel("Depth [mm
        ↪ ]");
    legend("Detected bubbles");
end

% COMPARE DETECTIONS
function comp = compare(res,auto,manu,tMinD,tMaxD,zMin,zMax)
    bubCount_all = res.bubCount_all;
    manCount = manu.manCount;
    tb = auto.tb;               tb_M = manu.tb;
    zb = auto.zb;               zb_M = manu.zb;
    amp = auto.AmpdB;           amp_M = manu.AmpdB;
    Ratio   = auto.Ratio_dB;    Ratio_M = manu.Ratio_dB;
```

```
    tCorrect = [];
    zCorrect = [];
    rCorrect = [];
    aCorrect = [];
    tFalse   = [];
    zFalse   = [];
    rFalse   = [];
    aFalse   = [];
    tMissed  = [];
    zMissed  = [];
    rMissed  = [];
    aMissed  = [];
    tChecked = [];
    zChecked = [];
    rChecked = [];
    aChecked = [];

    if (~isempty(tb))                   % exists detections
        if (~isempty(tb_M))             % exists manually counted
            for i = 1:length(tb)
                for j = 1:length(tb_M)
                    tDiff = abs(tb(i) - tb_M(j));
                    zDiff = abs(zb(i) - zb_M(j));
                    if (tDiff<tMinD) && (zDiff<zMin) && (~isnan(
                        ↪ tb_M(j))) && (~isnan(tb(i)))
                        tCorrect(end+1) = tb(i);
                        zCorrect(end+1) = zb(i);
                        rCorrect(end+1) = Ratio(i);
                        aCorrect(end+1) = amp(i);
                        tChecked(end+1) = tb_M(j);
                        zChecked(end+1) = zb_M(j);
                        rChecked(end+1) = Ratio_M(j);
                        aChecked(end+1) = amp_M(j);
                        tb_M(j) = NaN; tb(i) = NaN;
                        zb_M(j) = NaN; zb(i) = NaN;
                        amp_M(j) = NaN; amp(i) = NaN;
                        Ratio_M(j) = NaN; Ratio(i) = NaN;
                    end
                end
            end
            for i = 1:length(tb)
                for j = 1:length(tb_M)
                    tDiff = abs(tb(i) - tb_M(j));
                    zDiff = abs(zb(i) - zb_M(j));
                    if (tDiff>=(tMinD-0.1)) && (tDiff<tMaxD) && (
                        ↪ zDiff>=zMin-1) && (zDiff<zMax) && (~
                        ↪ isnan(tb_M(j))) && (~isnan(tb(i)))
                        tCorrect(end+1) = tb(i);
                        zCorrect(end+1) = zb(i);
                        rCorrect(end+1) = Ratio(i);
                        aCorrect(end+1) = amp(i);
```

```
                    tChecked(end+1) = tb_M(j);
                    zChecked(end+1) = zb_M(j);
                    rChecked(end+1) = Ratio_M(j);
                    aChecked(end+1) = amp_M(j);
                    tb_M(j) = NaN; tb(i) = NaN;
                    zb_M(j) = NaN; zb(i) = NaN;
                    amp_M(j) = NaN; amp(i) = NaN;
                    Ratio_M(j) = NaN; Ratio(i) = NaN;
                end
            end
        end
        tFalse  = tb(~isnan(tb));
        zFalse  = zb(~isnan(zb));
        rFalse  = Ratio(~isnan(Ratio));
        aFalse  = amp(~isnan(amp));
        tMissed = tb_M(~isnan(tb_M));
        zMissed = zb_M(~isnan(zb_M));
        rMissed = Ratio_M(~isnan(tb_M));
        aMissed = amp_M(~isnan(amp_M));
        disp(['** Detected: ', num2str(res.bubCount_all)]);
        disp(['** Manually detected: ', num2str(manCount)]);
        disp(['** Correct: ', num2str(length(tCorrect))]);
        disp(['** False: ',  num2str(length(tFalse))]);
        disp(['** Missed: ', num2str(length(tMissed))]);
    else                            % doesn't exists manually
        ↪ counted
        tFalse   = tb;
        zFalse   = zb;
        rFalse   = Ratio;
        aFalse   = amp;
        disp(['** Detected: ', num2str(res.bubCount_all)]);
        disp(['** Manually detected: ', num2str(manCount)]);
        disp('Only automatic detections, no manually.');
        disp(['** False: ',  num2str(length(tFalse))]);
        disp(['** Missed: ', num2str(length(tMissed))]);
    end
else                                % doesn't exist
    ↪ detections
    if (~isempty(tb_M))             % exists manually counted
        tMissed = tb_M;
        zMissed = zb_M;
        rMissed = Ratio_M;
        aMissed = amp_M;
        %disp('** Detected: 0 ');
        disp(['** Detected: ', num2str(res.bubCount_all)]);
        disp(['** Manually detected: ', num2str(manCount)]);
        disp('Only manual detections, no automatic.');
        disp(['** Missed: ', num2str(length(tMissed))]);
    else                            % doesn't exists manually
        ↪ counted
        disp(['** Detected: ', num2str(res.bubCount_all)]);
        disp(['** Manually detected: ', num2str(manCount)]);
```

```
            disp('No detections at all, both automatic and manual
                ↪ .');
            %disp('** Manually detected: 0');
        end
    end
    comp = struct('tCorrect',tCorrect,'zCorrect',zCorrect,'
        ↪ rCorrect',rCorrect);
    comp.tFalse = tFalse;   comp.zFalse = zFalse; comp.rFalse =
        ↪ rFalse;
    comp.tMissed = tMissed; comp.zMissed = zMissed; comp.rMissed
        ↪ = rMissed;
    comp.tChecked = tChecked; comp.zChecked = zChecked; comp.
        ↪ rChecked = rChecked;
    comp.aCorrect = aCorrect; comp.aChecked = aChecked; comp.
        ↪ aMissed = aMissed;
    comp.aFalse = aFalse;
end

function saveManual2Excel(rec,comp,EBRmin1)
    fileName1 = rec.fileName;  PathName = rec.PathName;
    tCorrect = (comp.tCorrect)';
    zCorrect = (comp.zCorrect)';
    rCorrect = (comp.rCorrect)';
    aCorrect = (comp.aCorrect)';
    rFalse   = (comp.rFalse);
    tFalse   = (comp.tFalse);
    zFalse   = (comp.zFalse);
    aFalse   = (comp.aFalse);
    tMissed  = (comp.tMissed);
    zMissed  = (comp.zMissed);
    rMissed  = (comp.rMissed);
    aMissed  = (comp.aMissed);
    rChecked = (comp.rChecked)';
    tChecked = (comp.tChecked)';
    zChecked = (comp.zChecked)';
    aChecked = (comp.aChecked)';

    pasNum1 = (str2num(PathName((end-1):end))); % vurdere
        ↪ bytte til str2double for raskere ytelse
    if (strfind(PathName,'Kat')~=0)
        type1 = 1;%'Kat4e_1/Pas';
    else
        type1 = 2;%'Kir4e_2/Pas';
    end
    fileLocStr='/Users/mytamlam/Dropbox/2021_MyLamProsjekt/Kode';
    fname = fullfile(fileLocStr,'
        ↪ OverviewManualDetections_test_w_all_manual.xlsx');

    if (~isempty(tCorrect))
        numC = length(tCorrect);
        corrL = ones(numC,1);
        type = type1.*corrL;
```

```
        pasNum = pasNum1.*corrL;
        fileName = (repmat({fileName1},numC,1));
        EBRmin = EBRmin1.*corrL;
        corrTable = table(type,pasNum,fileName,EBRmin,tCorrect,
            ↪ zCorrect,rCorrect);
        if (isfile(fname))
            coTable = readtable(fname, 'Sheet', 1);
            newCorrTable = [coTable;corrTable];
            writetable(newCorrTable, fname, 'Sheet', 1);
        else
            writetable(corrTable, fname, 'Sheet', 1);
        end
    end
    if (~isempty(tFalse))
        numF = length(tFalse);
        falsL = ones(numF,1);
        type = type1.*falsL;
        pasNum = pasNum1.*falsL;
        fileName = (repmat({fileName1},numF,1));
        EBRmin = EBRmin1.*falsL;
        falsTable = table(type,pasNum,fileName,EBRmin,tFalse,
            ↪ zFalse,rFalse);
        if (isfile(fname))
            fTable = readtable(fname, 'Sheet', 2);
            newFTable = [fTable;falsTable];
            writetable(newFTable, fname, 'Sheet', 2);
        else
            writetable(falsTable, fname, 'Sheet', 2);
        end
    end
    if (~isempty(tMissed))
        numM = length(tMissed);
        missL = ones(numM,1);
        type = type1.*missL;
        pasNum = pasNum1.*missL;
        fileName = (repmat({fileName1},numM,1));
        EBRmin = EBRmin1.*missL;
        missTable = table(type,pasNum,fileName,EBRmin,tMissed,
            ↪ zMissed,rMissed);
        if (isfile(fname))
            mTable = readtable(fname, 'Sheet',3);
            newMTable = [mTable;missTable];
            writetable(newMTable, fname, 'Sheet', 3);
        else
            writetable(missTable, fname, 'Sheet', 3);
        end
    end
    if (~isempty(tChecked))
        num = length(tChecked);
        checL = ones(num,1);
        type = type1.*checL;
        pasNum = pasNum1.*checL;
```

```
        fileName = (repmat({fileName1},num,1));
        EBRmin = EBRmin1.*checL;
        checTable = table(type,pasNum,fileName,EBRmin,tChecked,
            ↪ zChecked,rChecked);
        if (isfile(fname))
            chTable = readtable(fname, 'Sheet', 4);
            newChTable = [chTable;checTable];
            writetable(newChTable, fname, 'Sheet', 4);
        else
            writetable(checTable, fname, 'Sheet', 4);
        end
    end
end

% SAVE TO EXCEL
function saveRec2excel(rec, auto, TableBub)
    fileName = rec.fileName; PathName = rec.PathName;
    tb = auto.tb;
    sheetNum = 1;
    %switch EBRmin
    %    case 15
    %        sheetNum = 1;
    %    case 17.5
    %        sheetNum = 2;
    %    case 20
    %        sheetNum = 3;
    %    case 22.5
    %        sheetNum = 4;
    %    case 25
    %        sheetNum = 5;
    %end
    pasNum = int2str(str2num(PathName((end-1):end))); % vurdere
        ↪    bytte til str2double for raskere ytelse
    if (strfind(PathName,'Kat')~=0)
        type = 1;%'Kat4e_1/Pas';
    else
        type = 2;%'Kir4e_2/Pas';
    end
    %fileLocStr=['/Users/mytamlam/Dropbox/BubbleCountwRatio/',
        ↪ type, pasNum];
    fileLocStr='/Users/mytamlam/Dropbox/2021_MyLamProsjekt/Kode';
    %fname = fullfile(fileLocStr,[fileName(1:16),'Detec.xlsx']);
    fname = fullfile(fileLocStr,'
        ↪ OverviewDetections_test_w_all_manual.xlsx');

    if (~isempty(tb))
        if (isfile(fname))
            Table = readtable(fname, 'Sheet', 3);
            newTable = [Table;TableBub];
            writetable(newTable, fname, 'Sheet', 3);
        else
            writetable(TableBub, fname, 'Sheet', 3);
```

```
        end
    end
end

%function saveOverview2excel()

%end


% DRAW BUBBLES IN EARLYBIRD
function hBub = drawBubles(figh,hBub,tb,zb,marker)
    if ~isempty(hBub),delete(hBub);end
    CmAx = figh.UserData.hCm.Parent;
    hold(CmAx,'on');
    hBub = plot(CmAx,tb,zb,marker,'MarkerSize',12);
    hold(CmAx,'on');
end

% CONVERSION FROM DATENUM TO SECONDS
function t = dateNum2sec(tNum,rec)
    cmmode = rec.cmmode;
    t  = (tNum-datenum(cmmode.date_time))*60*60*24;
    %tDt=datetime(tNum,'ConvertFrom','datenum');
end

% CONVERSION FROM SECONDS TO DATENUM
function tNum = Sec2dateNum(t,rec)
    cmmode = rec.cmmode;
    p = rec.p;
    %t=(tNum-datenum(ob.acq_p.date_time))*60*60*24;
    tNum = datenum(cmmode.date_time)+ (t+p.t0)/(60*60*24);
end
```

A.2 THE ALGORITHM FOR MANUAL DETECTION

This section contains the algorithm for performing manual detections in the Color M-Mode image of the EarlyBird Software. This algorithm was used for the manual detections performed by Leth-Olsen et al. in his study on *Detection of Cerebral High Intensity Transient Signals by NeoDoppler During Cardiac Catheterization and Cardiac Surgery in Infants*. [13] The algorithm was developed by Hans Torp, and includes calculations of the position, amplitude and EBR of the manual detections performed. The algorithm is listed in Listing 7.

Listing 7: The Algorithm for Manual Detections

```
% bubleCount - script for manual buble marker
% 2020.03.25 Hans Torp
% 2020.03.29 Cloud markers included
% 2020.04.05 Uncertain markers, and Amplitude included
% 2020.04.08 bugfix in bubleAmplitudes
```

```
% 2020.04.14 bugfix; use string array in tables
% 2020.04.21 bugfix, prevent crash when clicking wrong panel+
    ↪ questdialog
% 2020.04.22 read back from excel and plot markers
% 2020.05.03 read back from excel bugfix
% 2020.10.01 marking bubles in right Cmmode allowed
% 2021.01.07 Update excel files with backgound amplitudes
% 2021.01.22 Update current excel file with buble amplitudes and
    ↪ background signals, with manual corrections
% 2021.09.11 Only single bubles

%% remove all buble marks. Run this first for new recording
if exist('hBub','var'),delete(hBub);end
hBub=[];tb=[];zb=[];

%% mark bubles in left Cmmode image; hit return to finnish
% zoom/ pan and repeat this to mark all bubles
[tb,zb,hBub]=markBubles(figh,hBub,tb,zb,'oy');

%% Write bublepos for single bubles to excel file
manualCor=1;
ob=guidata(figh);
fName=fullfile(ob.datapath,[ob.filename(1:16),'Hits.xlsx']);
if ~isempty(tb)
    TableBub=getSingleBublesTable(figh,tb,zb,manualCor);
    writetable(TableBub,fName,'Sheet',1);
end

%% Read back from excel and plot buble markers
% First remove all buble marks
if exist('hBub','var'),delete(hBub);end
hBub=[];tb=[];zb=[];

ob=guidata(figh);
fName=fullfile(ob.datapath,[ob.filename(1:16),'Hits.xlsx']);

myTableBub=readtable(fName,'Sheet',1);
if ~isempty(myTableBub)
    if isfield(table2struct(myTableBub),'SingleBubles')
        tb=myTableBub.tb;zb=myTableBub.zb;
        hBub=drawBubles(figh,hBub,tb,zb,'oy');
        tbSec=dateNum2sec(figh,tb)
    end
end

%% lokale funksjoner

function  TableBub=getSingleBublesTable(figh,tb,zb,manualCor)
ob=guidata(figh);
[AmpdB,Amplitude,Background_dB,tm,zm]=bubleAmplitudes(figh.
    ↪ UserData.Cmmode,tb,zb,manualCor);
tNum=datenum(ob.acq_p.date_time)+ tm/(60*60*24);
```

```
Ratio_dB=AmpdB-Background_dB;
SingleBubles=string(datestr(tb,'YYYY:mm:dd:HH:MM:SS:FFF'));
TableBub=table(SingleBubles,tb,zb,AmpdB,Amplitude,Background_dB,
    ↪ Ratio_dB);
end

function addBackGroundAllFiles(filename,manualCor)
%% Write bublepos, amplitudes and Background amplitude to all
    ↪ excel files in folder
if nargin>0
    searchStr=filename;
else
    searchStr='*.mat';
    manualCor=0;
end
excelStr='Hits.xlsx';
fldrInfo=dir(searchStr);
Nfiles=length(fldrInfo);
if Nfiles<1
    disp('No matching files found');
    return
end
for n=1:Nfiles
    if fldrInfo(n).bytes>5e3
        filename=fldrInfo(n).name;
        filenameExcel=[filename(1:16),excelStr];
        try
            Tbub=readtable(filenameExcel,'Sheet',1);
            fileOk=isfield(table2struct(Tbub),'SingleBubles');
        catch
            Tbub=[];
            fileOk=0;
        end
        if fileOk
            try
                S=load(filename,'Cmmode','p','acq_p');
                fileOk=isfield(S,'Cmmode');
            catch
                fileOk=0;
            end
            if fileOk
                tb=Tbub.tb;zb=Tbub.zb;
                [AmpdB,Amplitude,Background_dB,tm,zm]=
                    ↪ bubleAmplitudes(S.Cmmode,tb,zb,manualCor);
                Ratio_dB=AmpdB-Background_dB;
                SingleBubles=string(datestr(tb,'YYYY:mm:dd:HH:MM:
                    ↪ SS:FFF'));
                TableBub=table(SingleBubles,tb,zb,AmpdB,Amplitude
                    ↪ ,Background_dB,Ratio_dB);
                writetable(TableBub,filenameExcel,'Sheet',1);
                disp(['Updated file ',filenameExcel]);
            else
```

```
                disp(['Convert to new format: ',filename]);
            end

        end
    end
end
end

function [t2,z2,t,z]=bubleSignature(Cmmode,t1,z1)
B=0.01;
t=mean(t1);z=mean(z1);
dz=1e-3*(z1(2)-z1(1));
teta=atand(B/dz) %doppler angle
nz=find(1e3*Cmmode.depthAx>z,1);
nt=find(Cmmode.timeAx>t,1);
fi=double(Cmmode.fi(nz,nt))/127;
vz=-Cmmode.vNyquist/3/2*fi
Tt=B./(vz*tand(teta))
t2=t+[-Tt/2,Tt/2]
z2=z+1e3*[-dz/2,dz/2]
end

function [AdB,A,BackgrounddB,tm,zm]=bubleAmplitudes(Cmmode,tbub,z
    ↪ ,manualCor)
% Buble signal amplitude. Corrected for attenuation
%Background calc included
if nargin<4,manualCor=0;end
Taverage=10;
att=0.3;%dB/cm/MHz
f0=7.8;%MHz
att_mm=2*att/10*f0;
%convert t to seconds
t0=datenum(Cmmode.date_time);
t=(tbub-t0)*24*60*60;
tm=zeros(size(t));
zm=zeros(size(t));
AdB=zeros(size(t));
BackgrounddB=zeros(size(t));
A=zeros(size(t));
It=-5:5;
It=-20:20;
Iz=-1:1;
Nz=size(Cmmode.PdB,1);
if isfield(Cmmode,'dBStep'), dBStep=Cmmode.dBStep;else dBStep=1;
    ↪ end
dAx=1e3*Cmmode.depthAx;
dz=dAx(2)-dAx(1);
tAx=Cmmode.timeAx;
dt=tAx(2)-tAx(1);
Naverage=round(Taverage/dt/2);
for n=1:length(t)
    if t(n)>0
```

```
        nz=find(dAx>(z(n)-dz/2),1);
        nz=min(nz,Nz-1);
        nt=find(Cmmode.timeAx>t(n),1);
        Ppart=dBStep*double(Cmmode.PdB(nz+Iz,nt+It));
        [AdB(n),nMax]=max(Ppart(:));nMax=nMax-1;
        NNz=length(Iz);Nt=length(It);
        Itm=floor(nMax/NNz)+nt-floor(Nt/2);
        Izm=mod(nMax,NNz)+nz-floor(NNz/2);
        tm(n)=tAx(Itm);
        zm(n)=dAx(Izm);
        ind1=max(1,Itm-Naverage);
        ind2=min(size(Cmmode.PdB,2),Itm+Naverage);
        P=dBStep*double(Cmmode.PdB(Izm,ind1:ind2));
        Pm=median(P);
        IndL=find(P<Pm+2*std(P));
        BackgrounddB(n)=mean(P(IndL));
        if manualCor
            disp([num2str(n+1),' ',datestr(tbub(n))]);
            tP=tAx(ind1:ind2);tP=tP-mean(tP);
            figure(20);plot(tP,P,tP,0*P+BackgrounddB(n),tP(1)+
                ↪ Taverage/2,AdB(n),'*','LineWidth',2);
            disp('Mark buble peak; return to cancel');figure(20)
                ↪ ;[~,y]=ginput(1);
            if ~isempty(y)
                AdB(n)=y;
                plot(tP,P,tP,0*P+BackgrounddB(n),tP(1)+Taverage
                    ↪ /2,AdB(n),'*','LineWidth',2);
            end
            disp('Mark background; return to cancel');figure(20)
                ↪ ;[~,y]=ginput(1);
            if ~isempty(y)
                BackgrounddB(n)=y;
                plot(tP,P,tP,0*P+BackgrounddB(n),tP(1)+Taverage
                    ↪ /2,AdB(n),'*','LineWidth',2);
            end
            pause(1);
        end
    else
        AdB(n)=0;
        BackgrounddB(n);
    end
    A(n)=10^((AdB(n)+att_mm*z(n))/20);
end
end

function hBub=drawBubles(figh,hBub,tb,zb,marker)
if ~isempty(hBub),delete(hBub);end
CmAx=figh.UserData.hCm.Parent; % getcurrentaxis
hold(CmAx,'on');
hBub=plot(CmAx,tb,zb,marker,'MarkerSize',12);
hold(CmAx,'on');
end
```

```
function [tb,zb,hBub]=markBubles(figh,hBub,tb,zb,marker)
%% mark bubles in left Cmmode image; hit return to finish
% zoom/ pan and repeat this to mark all bubles
[tb1,zb1] = ginput;
if tb1(1)<200
    tb1=Sec2dateNum(figh,tb1);
end
t0=figh.UserData.hCm.XData(1);
t1=figh.UserData.hCm.XData(end);
zmin=figh.UserData.hCm.YData(2);
zmax=figh.UserData.hCm.YData(end-1);
coordOK=min(tb1)>t0 & max(tb1)<t1 & max(zb1)<zmax & min(zb1)>zmin
    ↪ ;
if ~coordOK
    errordlg('One marker outside Color Mmode boudaries');
    return;
end
tb0=tb;zb0=zb;
tb=[tb;tb1];zb=[zb;zb1];
hBub=drawBubles(figh,hBub,tb,zb,marker);
SpAx=figh.UserData.hSp.Parent;
pan(SpAx,'xon');
answer = questdlg('Add new bubble markers?','','Yes','No','Yes');
if strcmp(answer,'No')
    tb=tb0;zb=zb0;
    hBub=drawBubles(figh,hBub,tb,zb,marker);
end
end

function t=dateNum2sec(figh,tNum)
%convert timeaxis tNum from detenum to seconds
ob=guidata(figh);
t=(tNum-datenum(ob.acq_p.date_time))*60*60*24;
%tDt=datetime(tNum,'ConvertFrom','datenum');
end

function tNum=Sec2dateNum(figh,t)
%convert timeaxis tNum from detenum to seconds
ob=guidata(figh);
%t=(tNum-datenum(ob.acq_p.date_time))*60*60*24;
tNum=datenum(ob.acq_p.date_time)+ (t+ob.p.t0)/(60*60*24);
end
```

My Tam Lam

Automatic Detection of Air Emboli in The Cerebral Circulation in Newborns By Ultrasound Doppler Technique

NTNU
Norwegian University of
Science and Technology