

MASTER'S THESIS

Thesis submitted in fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Medical Engineering & eHealth

Time-variant tomographic reconstruction from ECG

By: Bc. Václav Indrák
Student Number: 2310228023

Supervisor 1: FH-Prof. Dr. Andreas Drauschke
Supervisor 2: Ing. Jiří Sekora, MBA

Vienna, 22. May 2024

Declaration of Authenticity

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz/ Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and according to the rules currently applicable at the UAS Technikum Wien and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.”

Brno, 22. May 2024

Place, Date

Digital Signature

Kurzfassung

In dieser masterarbeit wird ein entwurf für ein mögliches neues nicht-invasives diagnosegerät vorgeschlagen, das auf elektrische pathologien des herzens abzielt und mehrleitungs-EKG-aufzeichnungen und einen auf tomographischer rekonstruktion basierenden algorithmus verwendet. Zunächst werden die notwendigen schritte für diesen entwurf dargelegt. Außerdem wird eine kurze theorie zu allen verwendeten ansätzen sowie Informationen zu den derzeit verwendeten methoden gegeben. Anschließend wird das gesamte system detailliert beschrieben und es werden vielversprechende ergebnisse an verschiedenen realen datensätzen gezeigt. Die gesamte arbeit dient als Proof of Concept für ein komplexeres zukünftiges design.

Schlagwörter: EKG, Tomographische rekonstruktion, Nicht-invasive diagnose

Abstract

This master's thesis proposes a design of possible new non-invasive diagnostic device aiming for electrical pathologies of the heart, utilizing multi-lead ECG recordings and tomographic reconstruction-based algorithm. First, necessary steps for this design are laid out. Also, a brief theory is provided for all the used approaches along with information about currently used methods. Then a detailed description of the whole system is provided, later showing promising results on different sets of real data. This entire work serves as a proof of concept for a more complex future design.

Keywords: ECG, Tomographic reconstruction, non-invasive diagnosis

Acknowledgements

I would like to thank to my supervisor, Jiří Sekora, for all the help he gave me during development of this project, and advice regarding current and future work. I also would like to thank to Brno's University of Technology for allowing and supporting this project. Lastly, I would like to thank Professor Andreas Drauschke for help with structure and contents of this thesis.

Table of Contents

1	Introduction.....	6
2	Theoretical background.....	7
2.1	Medical background	7
2.2	Now used methods.....	8
2.2.1	ECG.....	8
2.2.2	Intracardial potential mapping.....	10
2.3	Hardware	10
2.4	Software	12
2.4.1	Linear filters	13
2.4.2	Wavelet transformation filters	13
2.4.3	AI powered tools.....	15
2.4.4	Tomographic reconstruction	16
3	Main work	18
3.1	Hardware	18
3.1.1.1	First prototype	19
3.1.1.2	First part	20
3.1.1.3	Second part	21
3.1.1.4	Extra circuit parts.....	22
3.1.1.5	Data acquisition.....	23
3.1.2	Model improvements.....	24
3.2	Data preprocessing	24
3.2.1	First prototype	25
3.2.1.1	Serial communication and data reconstruction.....	25
3.2.1.2	Filtering and real data creation	26
3.2.1.3	Data sorting and vector creation	26
3.2.2	Improvements.....	27
3.2.2.1	Filtration.....	27
3.3	Reconstruction algorithm	28
3.3.1	First prototype	28
3.3.1.1	Reconstruction	29

3.3.2	Improvements.....	31
3.4	Visualisation of results	32
3.4.1	Greyscale visualisation	32
3.4.2	RGB visualisation	32
3.4.3	Vector visualisation	33
4	Results.....	34
5	Discussion	42
	Bibliography.....	44
	List of Figures.....	46
	List of Abbreviations	47
	A: Code for Arduino.....	48
	B: Created measuring circuit.....	50
	C: Recorded data	51
	D: Python code.....	52

1 Introduction

This thesis proposes a device concept, that would allow non-invasive diagnosis of heart conditions in terms of its electrical activity, while also being easier to interpret.

In a current practice, diseases of the heart are becoming increasingly more common, as well as their treatments are becoming more effective. One of the key parts in diagnosing and treating such conditions is its early detection. In modern medicine exist many ways of diagnosing heart condition providing very good results, however most of them have certain drawbacks. This thesis proposes a proof of concept of a new design for a device, that could improve diagnosing conditions linked with electrophysiological changes.

The proposed device would be a step between a standard twelve-lead ECG and full reconstruction acquired from intracardial potential mapping, therefore giving more viable and simplified information without the need for invasive and expensive procedure or extensive training in reading of ECG.

The proposed design utilizes tomographic reconstruction, creating a time-variant 2D and in future development even 3D model of the heart's electrical activity. Main source of measured data is an electrode field on the chest measuring a bipolar ECG of all electrode pairs.

The main goal and idea behind this device is to present more information on the heart than a standard ECG would, the results being much easier to understand even by personnel untrained in deep cardiology and providing much higher accuracy on all the detectable ailments, such as more precise location and size of the area affected by infarction or sources and nature of arrhythmias. In theory, an infarction would be visible on the resulting model as an empty inactive space in a place, where activity should occur. Arrhythmias would then be visible due to specific behaviour of the action potential, such as different point of origin for excitation or different excitation paths.

In general, this device could then be used in hospitals during first examinations instead of standard ECG as it would provide simpler and more detailed information about the heart, or by cardiologists looking for more details regarding the hearts condition.

The device created in this work is a proof of concept and nowhere near being ready for any practical use, but successful prototype will open further development and improvements.

2 Theoretical background

2.1 Medical background

Heart works as the main source of transportation for everything in the body. Operating as almost perpetual pump, it pushes the blood through arteries, ensuring that all nutrients and waste products are transported to where they are supposed to be. Most of the entire organ is composed of conductive contractable muscle tissue that is unique in the body.

The whole electric activity is a semi-independent system, that can operate even when severed from the neural control of the body thanks to the conductive system. The entire pacemaking system is well organised with several failsafe systems in case of failure of a previous part. Pacemaking starts in the sinoatrial node, where influences from other control mechanisms meet and the heartrate is decided. From there, the depolarisation, therefore electric activity is transferred through dedicated fibres onto both atria, resulting in its spreading across atria and their followed depolarisation and contraction. All this electric activity is then collected in atrio-ventricular node. From there, with some delay, it continues to bundle of His where a delay is created, so atria and ventricles don't contract too closely. Then the signal is separated to left and right, going through respective bundle branches to the tip of the heart and onto the Purkyně fibres, which facilitate the following depolarisation and contraction of the ventricles. All this is then later followed by repolarisation waves, preparing the myocard for the next round.

The whole conductive system of the heart is very robust, but also crucial to correct functionality of the heart. Unfortunately, the electric responses of the conductive system alone are so small that its detection is almost impossible. What is then measured as electric activity of the heart, usually by ECG, is the depolarisation and repolarization of the myocard. In ECG, depolarisation of atria is represented by the P wave, its repolarisation is then hidden in the QRS complex, that represents depolarisation of ventricles. T wave is then repolarisation of ventricles.

In all cases, the actual mechanical activity of the heart is delayed to the electrical one as the complex mechanisms of the myocard take some time to transform the electrical excitation into contraction.

One very interesting and important mechanism of myocard is its protection against tetanic spasm. Normal muscle tissue has a refractory period after contraction very short and the excitations can even stack up, leading to so called tetanic spasm, where the muscle is contracted the whole time, when excitation is present. For heart, this could be very dangerous and cannot happen. The defence mechanism for this is very long latency period after excitation, during which nothing is happening with the cells and as they are depolarised, they are immune to all new incoming stimuli. This effectively prevents the myocard cells from contracting multiple times during one heart cycle. This effect is also visible on ECG as the delay between QRS complex (depolarisation of atria) and T wave (repolarisation of atria). [1]

2.2 Now used methods

Modern diagnostic methods differ in many of the basic principles of functionality but can be generally divided into two categories. First being anatomical methods used for visualization such as ultrasound, CT or MRI, the second being functional methods dealing with physiological state of the heart (mentioned later). Most from the second category deal with electrical activity of the heart, which will also be the only discussed here.

2.2.1 ECG

Most common and simple way of diagnosing a heart is ECG. It is widely used across many different fields in hospitals, having several different variations of main measurement. A great strength of ECG comes from its simplicity, where with just three electrodes, you can diagnose many different conditions rather effectively. It also has rather low costs. The main hardware and software for measuring ECG is rather simple. But this comes with a great disadvantage, being the difficulty of recognizing and finding nonacute persisting issues. To fully recognise everything possible from an ECG recording, doctors need to study and train for many years and still can make mistakes. It's not a rare occurrence for a cardiologist to be called into a clinic, because the current doctor cannot understand something out of the ordinary.

In general, basic ECG used is simple three electrode recording, where one of the electrodes is not even measuring own signals but acting as an agent improving the actual measured data by lessening the effects of 50 Hz main grid interference or the base line drift. It's called "driven leg". This is visible on Figure 1, where driven leg line leaves AD705J amplifier and goes onto the right leg. The other two electrodes then measure the actual ECG as a bipolar recording, usually between arms or specific places on the chest. This method is simple, but much less effective, as many things can only be detected, but not specifically annotated. Therefore, it's mostly used in cases, where simplicity is either required or just enough, such as HRV measurements or ICU monitors.

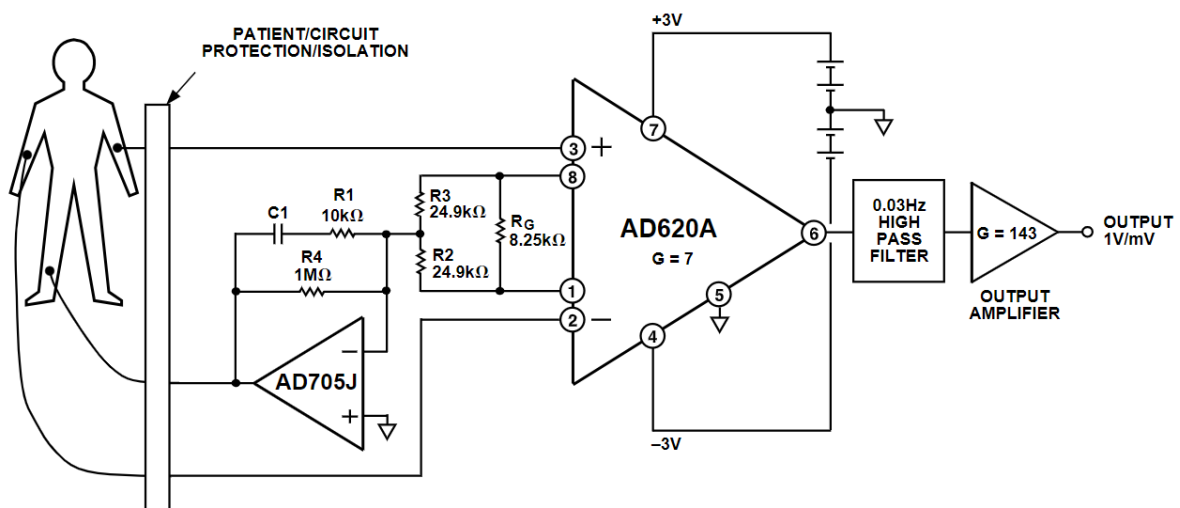


Figure 1: Example of simple ECG measuring schematic from AD620 datasheet [2]

The more detailed twelve-lead recordings are more complicated and specific versions of the previous. These are used for qualitative analysis of the ECG, as they show the electric activity from different angles and therefore contain more information. As they yield more data, they are more dependent on the accuracy of the measurement, which explains why this method is not used more often instead of the three-lead system. Any movement of the electrodes introduces new motion artifacts into the measurement, which in this case is very difficult to remove. For this, the twelve-lead system is not continuously used on patients in bed as these artifacts would occur constantly.

The whole twelve lead system consists of three independently used systems combined. In total, six of the recordings are of a bipolar nature, the other six are unipolar. Because of certain overlap in some of the recordings, only ten electrodes are used for the measuring, one of them being the afore mentioned “driven leg”. Of the bipolar electrodes all three are placed on the extremities, one on each arm and one on left leg. The non-measuring electrode is placed on the right leg, as it’s the furthest part of body from the heart. These four electrodes can either be placed on the end of the limbs (usually ankles and wrist) or can be placed along the first joint of the limb (shoulders and hips).

The first of the original systems are Einthoven’s leads, that form the Einthoven’s triangle. These leads use standard potential difference between the limb electrodes, giving leads I, II and III, where I is measured between right and left arm, II between right arm and left leg and III between left arm and left leg.

Second of the bipolar original systems are Goldberg’s leads. This system is directly dependant on the Einthoven’s system, as it uses the same original electrodes and calculates the resulting measurement from them. All these recording use one of the three electrodes as “exploring electrode” or as the positive one and as reference uses the centre of the other two. Resulting leads are aVR with right arm as the exploring electrode, aVL with left arm as exploring and aVF with left leg as exploring. In many cases the resulting values are inverted for different appearances and values. As these are derived from the Einthoven’s system, they are also directly calculated from the recordings by following equations:

Equations for Goldbergs leads 1 [3]:

$$\begin{aligned}
 -aVR &= \frac{\text{Lead I} + \text{Lead II}}{2} \\
 aVL &= \frac{\text{Lead I} - \text{Lead III}}{2} \\
 aVF &= \frac{\text{Lead II} + \text{Lead III}}{2}
 \end{aligned}
 \tag{1}$$

The last six chest electrodes are measured as unipolar, meaning that all of them use a common centre as a reference. This point is called Wilson’s central terminal and is calculated as a centre of the Einthoven’s triangle, therefore uses average of the three limb electrodes. These

chest leads then result in V1-V6 recordings, that hold new angles and information about the heart.

One of the best advantages of ECG is the simplicity of acquisition. It is very simple, fast and non-invasive, therefore it can be measured at almost any conditions with very little expenses for separate measurements. However, a deep qualitative analysis can be challenging and still requires trained specialists to perform. However, many new automated ways of delineation are being developed. [3]

2.2.2 Intracardial potential mapping

One of the most advanced methods for hearts diagnosis, and often for treatment as well, is reconstruction of intracardial potentials - electrophysiological procedure. This method binds together very accurate mapping of electrical activity in desired parts of the heart with anatomical properties of the same part. This allows for the most accurate diagnosis of many conditions altering the hearts conductivity. Unlike previously mentioned ECG, this procedure is invasive and requires very specialised and expensive equipment, along with specially trained personnel.

For this procedure to happen, special catheters need to be placed in the heart, usually accessing through the femoral vein. In the heart, several control electrodes are placed, mostly near the point of depolarization start, these act as reference electrodes. Then the special catheters come to work. Their real-time position is tracked at all times using either electromagnetic or impedance navigation. This allows for the anatomic reconstruction of the part of the heart, onto which the electric responses are mapped. The electric mapping can work on different modes such as time dependant, where the model shows the time needed for the signal to reach from the reference electrode to the current position, or impedance, where the impedance of myocard currently in contact is measure and displayed as conductivity. This allows for many types of arrhythmias to be visible, such as points for re-entry in atrial fibrillation. The catheters used consist of long working area with multiple electrodes shaped into spiral or star. This allows for much larger coverage of measuring and also for the impedance mapping. Based on the resulting model, therapeutic ablation is usually performed, where specific places after ablation loose their conductive ability, preventing for example re-entries in atrial fibrillation. [4], [5]

2.3 Hardware

When measuring ECG, hardware can be the key to success of high-quality recording. Everything along the path of the signal needs to be taken into account and finetuned for optimal results. The general hardware itself is not as complicated, but its clinical variants are.

Everything starts with electrodes. Most commonly used ones now are single-use silver-chloride ones, mostly differing in the quality and strength of the adhesant. These electrodes combine good impedance response with great robustness and simplicity, meaning they can be placed

on the skin with very little preparation as cleaning and usage of ECG gel. Due to their design, they are also rather durable and can, without bigger problems, withstand harsh use even during extensive movement. These are the most used ones in general practice, even with their disadvantage of not being reusable, meaning every measurement costs some money.

Other alternative can be solid metal electrodes with clips or suction balloons, that rely solely on own conductivity of skin. These electrodes often have a bit better impedance response, but that is heavily outweighed by their higher difficulty of use. These days, metal electrodes are mostly used for educational purposes, as they show the influences of factors on the measurement more and are cheaper to use. [6]

Another important part of the hardware, even though often overlooked, are wires and cables. Standard cables used for computer connections are already perfected, but the ones used for electrode connection or inter-circuit connection are often not. In laboratory conditions (in schools mostly) standard insulated copper wires are used, but for higher performance and in practice, special shielded wires are used, as they prevent parasitic interferences from the surrounding on the main signal.

The main electrotechnical parts for ECG measurement consist of operational amplifiers, resistors and capacitors. Amplifiers serve several purposes. They provide impedance insulation from the body, which is important as body is very weak source of voltage, therefore any strain on the signal would affect it greatly. They also serve as the main comparative block to output the actual voltage created by potential differences between the electrodes. Lastly, they also amplify the measured signal, so its values can be measured in acceptable units. Many requirements are placed on amplifiers, and they even differ based on the part of the circuit. Some of the amplifiers used are general ones, that can be found in many other nonmedical areas, other ones are specifically designed for use in medical devices, as they boast with the best parameters needed for this application. One of those parameters is Common Mode Rejection Ratio (CMRR), that for medical amplifiers is much larger than for other uses. An example of amplifier, in this case instrumentation amplifier, designed mostly for use in medical applications is AD620 by Analog Devices, that even shows a diagram for ECG measurement in its datasheet (shown in Figure 1).[2], [7]

Resistors are used to set the amplification of the amplifiers and with capacitors are the main building blocks for all hardware signal filtration, whether used as simple T-blocks or as active filters with use of amplifiers [8]. Mostly used filters are lowpass antialiasing filters ensuring the Nyquist theorem of sampling frequency and highpass filters removing the baseline drift. In modern use, most of the filtration is done digitally. Resistors are also one of the reasons for higher costs of professional equipment, because the ones used in practice need to be very precise in their resistivity, which is difficult and expensive to achieve.

Another electrical part widely used in ECG hardware is diode, especially Zenner's, as it can serve as very simple surge protection. Zenner's diode has a very distinctive feature, where in positive direction of the current flow it acts as simple conductor, only altering the current of voltage slightly due to its inner semiconductive mechanisms. More important is the negative conductivity, where up to a certain voltage (determined by how it's made) it acts as an insulator

not allowing anything to pass, but when the break-through voltage arrives, it changes to a conductor with very little resistivity, allowing a surge to go outside of the circuit. [9]

Last important part for measuring hardware is analog-digital converter. This part converts the infinite possibility values of analog signal into its rounded digital form that can be processed by computer. Type and quality of the converter are also very important as it needs to be fast enough to handle desired sampling frequencies without causing interferences and have sufficient bit depth to provide good range of values of the measured data. The voltage span is also important, as it also determines the possible measured values. Mostly used converters have 10- or 12-bit depth, meaning the resulting values can go from zero to 1024 for 10 bits or 4096 for 12 bits. When voltage span is taken into account, a step value can be calculated showing how much voltage increase is needed for the measured value to increase by 1. For example, 10-bit converter with voltage span of 0-5 V has a step of 4,88 mV per value.

Another part worth mentioning is galvanic separation. As mentioned before, all ECG devices need to have very good surge protection to prevent any harm being done to the patient in case of main electric grid failure or a failure of inner circuits. One part of this is aforementioned diodes, the other part is galvanic separation. This means that most of the measuring hardware, including the processing computer, are not directly electrically linked to the patient. For power sources, special types of transformers or separators are used, or the whole device can be battery powered. For the dataline, methods of complete electrical separation are used, such as wireless transmission in more advanced devices or utilization of optical data transfers (either optical cable or separators).

2.4 Software

Nowadays in ECG recordings the software is usually the most important and distinctive part of any device. Digital data filtering has almost completely replaced hardware filters, all visualised data are digitalised and processed before being shown. This all is thanks to the advancements in data and signal processing.

Leaving aside the parts of the data being transferred into the computer, as that is similar in many computer powered fields, the first and most important software part is filtration. In some uses, it's only a part of the preprocessing of the data, in standard ECG devices it's the core of it all. Digital signal processing has some history behind it and is a constantly evolving field. Methods now used are no longer similar to the original hardware filters, which often makes them more potent, as they don't have physical limitations. Previously, linear filters derived from hardware were used, both IIR and FIR posing certain advantages and disadvantages above each other, but they still keep their main strength - simplicity. They were replaced by currently most used methods utilizing the wavelet transformation, a nonlinear mathematical operation boasting with very high effectivity and usability, while still usable in real-time, but coming with higher complexity both for design and computing.

2.4.1 Linear filters

Linear filters as FIR (Finite Impulse Response) or IIR (Infinite Impulse Response) are, as mentioned before, digital derivatives from original hardware filters. The main difference between the two is apparent from their name. Impulse response is created by sending a Dirac's impulse (a single value of one followed by zeros) through the filter. With FIR filters, this impulse response has the same length as number of poles (or number of elements of the filter), as the values used for calculation are only taken from the input impulse, meaning when only zeros are coming in, only zeros can be coming out. Opposed to that, IIR filters will never stabilize to zero, as they use previously calculated values to calculate the following one. This means that whenever an IIR filter has a non-zero result, it will keep giving non-zero results until stopped. Both these filters come with certain advantages and disadvantages. The FIR filters are easier to create and have linear phase delay, meaning that they don't affect the morphology of the signal, but they need much higher complexity to achieve the same filtering results as IIR. Those have, as mentioned, much better filtering abilities, meaning they can filter much smaller band with same complexity. On the other hand, they have nonlinear phase shift, so the resulting signal can be altered in morphology. This can be an issue with real-time applications, but for offline applications was already dealt with, by applying the same filter twice, once forwards and once backwards, which essentially eliminates the phase shift and also doubles the filtration effect. [10]

2.4.2 Wavelet transformation filters

As mentioned before, the currently most used methods are based on wavelet transformation. These approaches utilize wavelet transformation (its discrete variant), an algorithm essentially composed of series of filters, creating multiple layers of the original signal only containing certain frequencies. Because of that, many different things are better visible and removable. This approach, however, works best for broad spectrum interferences and is not that good for removing standalone frequencies such as 50 Hz main grid interference. For better understanding, the wavelet transformations at each level splits the spectral contents of the signal in half, meaning that for a signal with sampling frequency of 2000 Hz, the first level creates two signals, one with spectral contents of 0-500 Hz and second with 500-1000 Hz (this only takes into account half of the spectra due to Nyquist theorem). In deeper levels, this is repeated, but only with lower frequency band, meaning that the same signal decomposed to fourth level would have five separate "signals", with spectral contents of 0-62,5 Hz, 62,5-125 Hz, 125-250 Hz, 250-500 Hz and 500-1000 Hz. Differences are then in the used method of the transformation, whether it's DTWT (Discrete Time Wavelet Transformation) that uses downsampling for creation of lower frequency bands, resulting in shorter signal at each level, or SWT (Stationary Wavelet Transformation) that maintains the length of the signal. For these decompositions, different wavelet families can be used, each of them having slightly different shape and providing slightly different results. For each family a different lengths can

also be used that again can provide different results. However, some wavelets from some families can be identical to ones from other families such as sym2 is identical to db2.

Basic filtering with this method is theoretically not difficult, but it contains several decisions and for some of them a clear empirical answer does not exist, therefore we must try and then decide. Among these decisions is which transformation algorithm to use – DTWT or SWT. In this case, for ECG the answer is suggested in [11] and semi proved, so for filtration the better approach is SWT. Other decisions, such as which wavelet to use or how to set the coefficient multiplier, are mostly trial and error.

The filtration itself is mostly done by altering the coefficients of the transform (the resulting signals), either by some forms of filtration or mostly by thresholding. Thresholding itself is not that difficult but getting the threshold is very tricky. One of the ways is standard deviation estimation using median, as high peaks of QRS complexes don't really influence it.

Equation for standard deviation estimation 2 [11]:

$$\sigma = \frac{\text{Median}|y_m|}{0,6745} \quad (2)$$

Once the threshold is set (different for each level of decomposition) everything below this threshold on each level is set to zero (except the lowest level, as it contains mostly only ECG and altering would damage it a lot). An example of signal decomposition with respective threshold is shown in Figure 2. Another question is the manner of thresholding, meaning how to handle values above the threshold. One option is hard thresholding, which basically leaves everything higher than the threshold as it was, but this can create disturbances as a sudden step would appear for any wave. Other options are soft or Garrote thresholding, each presenting new ways of handling this, but each bringing different downsides.

Once all the coefficients are changed accordingly, an inverse reconstruction is then performed, resulting in the filtered signal.

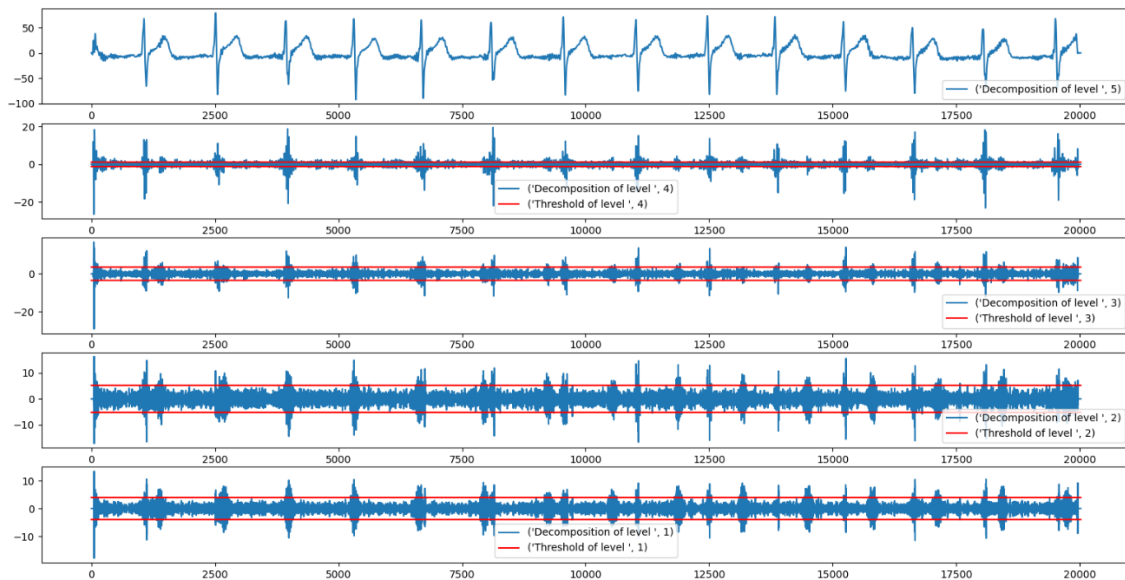


Figure 2: Stationary wavelet decomposition with thresholds for each level

Among other things, wavelet transformation is also very useful for detection algorithms, not only for ECG but for other signals or fields as well. This is thanks to its spectral-time dependant decomposition, allowing to search for different thing in several layers, while they remain in the same place. [11], [12]

2.4.3 AI powered tools

The current most developed and tested approach is, as in many other fields, machine learning or artificial intelligence. In most cases, for preprocessing are used different modified versions of convolutional networks. This, however, has a lot of limitations, mostly in data dependency and frequency specificity, therefore it's not widely used yet. [13]

Where AI and machine learning finds a great field of use are different types of algorithms, that aim to extract additional information or automate annotation. One of greatly used AI approaches is for automated segmentation of all components of ECG, such as marking very precisely beginnings and ends of certain complexes. Different approaches were used in the past, but compared to AI powered systems were very complicated and demanding to run with issues with certain episodes or states in recordings. On the other hand, AI powered methods have issues of being very data specific, meaning that some things will not be detected or will be detected incorrectly if the learning dataset doesn't contain certain instances. Also, a lot of systems have a set sampling frequency, that is very hard to change.

Another type of software used in cardiology are automated systems for computing useful parameters from recordings. Part of these systems are usually previously mentioned automated detections of certain elements in the signal, that are then used for the calculations. An example could be program automatically detecting R waves in QRS complexes of long-

term holter recordings, that are then used to calculate parameters for HRV. This is currently mostly done with wavelet transformation, but many different approaches are being tested utilizing different types of machine learning methods.

In research, many other programs are created and used to determine different conditions, that were not diagnosed from ECG before.

2.4.4 Tomographic reconstruction

Tomographic reconstruction is a process based originally on a mathematical function Radon transform, introduced by Johann Radon, Austrian mathematician, in 1917 [14], more specifically on its inverse part. Back then, the applications of it were not as feasible and clear as they are today. This function essentially transforms an object of certain dimensionality into a series of projections of lower dimensionality. Most common example is a sinogram, a series of 1D vectors created from a 2D image, example visible in Figure 3. The resulting vectors, or projections are each a sum (or integral in non-discrete applications) of the original model along one axis each time rotated by a certain degree. The resulting sinogram therefore has on one axis dimensions of the studied model, the other axis are degrees of rotation, usually going from zero to 180. A full rotation is almost never used as the two halves are identical.

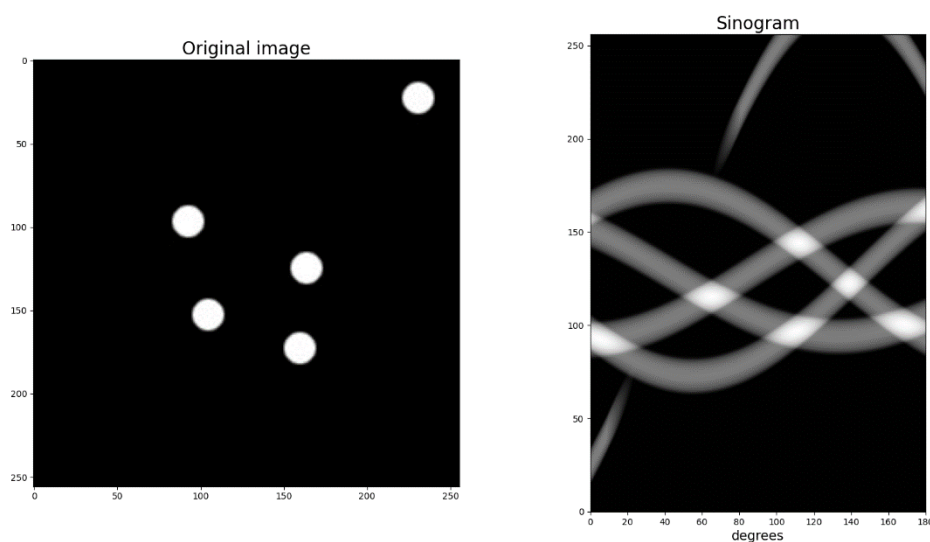


Figure 3: Example of picture decomposed to sinogram

The inverse transformation then takes the sinogram and reconstructs the original model from it. The basic principle is that the vector of values is “smeared” over the reconstructed space under corresponding angle from which it was measured or created. This approach is simple, but has a lot of negatives as the resulting model is never identical to the original one. One thing is presence of background noise that occurs due to the reconstruction, as the positive values

of the reconstructed object are added even to the background. Other thing is much lower sharpness of the edges, or even missing parts in the corner of square models or star artifacts. Some of these issues (such as background and edges) are targeted by using filtered back projection. For this, each vector of the sinogram is filtered by special filter that increases steepness of edges and also introduces negative values around the edges to negate the leftover background values from other projections. However this does not mitigate all the possible issues, such as star artifacts or missing corner objects.

The whole tomographic reconstruction based on Radon's transform has different approaches for calculation. In modern systems two main approaches are used: algebraic space reconstruction or Fourier slice reconstruction. Each of these approaches has certain points to it, but also brings some disadvantages. For example, Fourier-based algorithms have better edge steepness, but are more demanding for computing. For this reason, hybrid systems are sometimes created allowing the user to choose, which approach will be used. [15]

New innovations and improvements in tomographic algorithms now more focus on better optimizing the resulting model, rather than creating new mathematical methods, as the current ones are very powerful already. The most common addition in these algorithms lately are iterative improvements. This simply means that the first model created by the reconstruction is iteratively adjusted based on some form of error margin. Mostly used error value is created by calculating new projections from the constructed model and comparing them to the original ones. The adjustments then take different forms, from constructing error model from the error vectors or by subtracting the error values from the model directly.

In general, methods for tomographic reconstruction are being improved constantly, as they can be used in many different areas, not only medical technology. From devices utilizing these algorithms are worth mentioning CT scanners (Computed Tomography), many approaches used in MRI (Magnetic Resonance Imaging), or from non-medical field electron microscopy. [15]

3 Main work

The device created in this work is a proof of concept and nowhere near being ready for any practical use, but successful prototype opens options for further development and improvements.

However, due to the complexity of such design and its demand for resources, a proof-of-concept approach is chosen, creating a simpler design utilizing lower quality means to prove, whether this idea is feasible.

With the potential goals of future development, measuring all the recordings right away would not be very feasible. Instead, values nearing each electrodes potential are measured, allowing the bipolar recordings to be created digitally, alleviating the high demands on the measuring hardware.

All the recordings will then be pre-processed in terms of their quality (filtration of all interferences) and their sorting for easier run of the reconstruction algorithm. The core reconstruction will then happen with modified version of tomographic reconstruction utilizing a sample-by-sample approach. From each time sample (or a short series of them) will then a single 2D reconstruction be created. Those will then be connected into a 3D matrix, creating a time-variant model displaying dynamic changes of the heart's electrical activity.

To complete a design like this, several steps need to be taken. First step is to create measuring hardware, as no similar is publicly available for measuring required type of data. As a prototype concept, all is made with regards to affordability combined with sufficient results. This includes building the hardware measuring system as well as ensuring correct loading of the data for later processing.

Second part is preprocessing, which contains several steps. All the measured data needs to be converted to bipolar ECG recordings, all the recordings need to be then filtered and sorted to ensure the main algorithm runs without setbacks and with good accuracy.

Next part is the core reconstruction algorithm. As this is an experimental approach, a lot of trial and error is needed.

Last part is then testing the whole to understand the real results and if it works, what are the drawbacks and what can be improved.

As this work is a proof of concept, multiple improvements will be added gradually, even outside of this thesis, in hopes of creating better results and understanding of potential capabilities.

3.1 Hardware

For measuring hardware, several things are needed. First thing are electrodes for measuring. As this is an experimental work, single use electrodes will be used, as they are very simple and easy to use. Then the main measuring of the electrodes potentials takes place. Unlike in standard ECG, the recorded data is not the final bipolar ECG, which makes the measuring more challenging. Reason behind that is, that potential final prototype would be using several times more electrodes, producing even more ECG recordings, which would be almost

impossible or extremely expensive and complex (for example a prototype measuring 256 electrodes would produce over 50000 unique signals). The challenge lies with measuring a single potential, as standard ECG recordings are already measured as potential difference, therefore a pure voltage. An idea for measuring not precisely the potentials, but values close to it, was created, where all the potentials would be measured against a common centre created by combining all the measured signal together. This would ensure theoretical feasibility of the values, while also providing a reference with common phase interferences, such as 50 Hz main grid.

This common centre also provided the base for creating the “driven leg”, as the measured data were not acceptable without it and its use improved the results significantly.

Another issue is the main analog-digital conversion. With use of high sampling frequencies, the data measurement and its transfer is becoming an issue, especially with regards to cost requirements. For this, already existing microchip boards will be used along with their AD-converters. This solution creates a small delay between measured electrodes and needs to be modified in terms of inner drivers and commands to handle the dataflow, but overall is a decent substitute for creating everything anew.

3.1.1.1 First prototype

The first prototype was created in regard to testing the main idea and simple constitution. Therefore, all of the hardware was made in an electrotechnical laboratory on a circuit board using larger components with lower value precision for their affordability. This was done so it could be taken apart in the end and almost no resources would be spent. Firstly, the circuit layout was created using Autodesk Fusion program (or Eagle in older versions), that enables modelling and even simulations of circuits. All the necessary values of components and their placement were chosen along with specific types of some parts, such as amplifiers. During construction of the hardware, several shortcomings were encountered and forced changes in the original circuit design. For example, values of resistors following the first amplifiers leading to the common centre were changed four times, while testing seven different values.

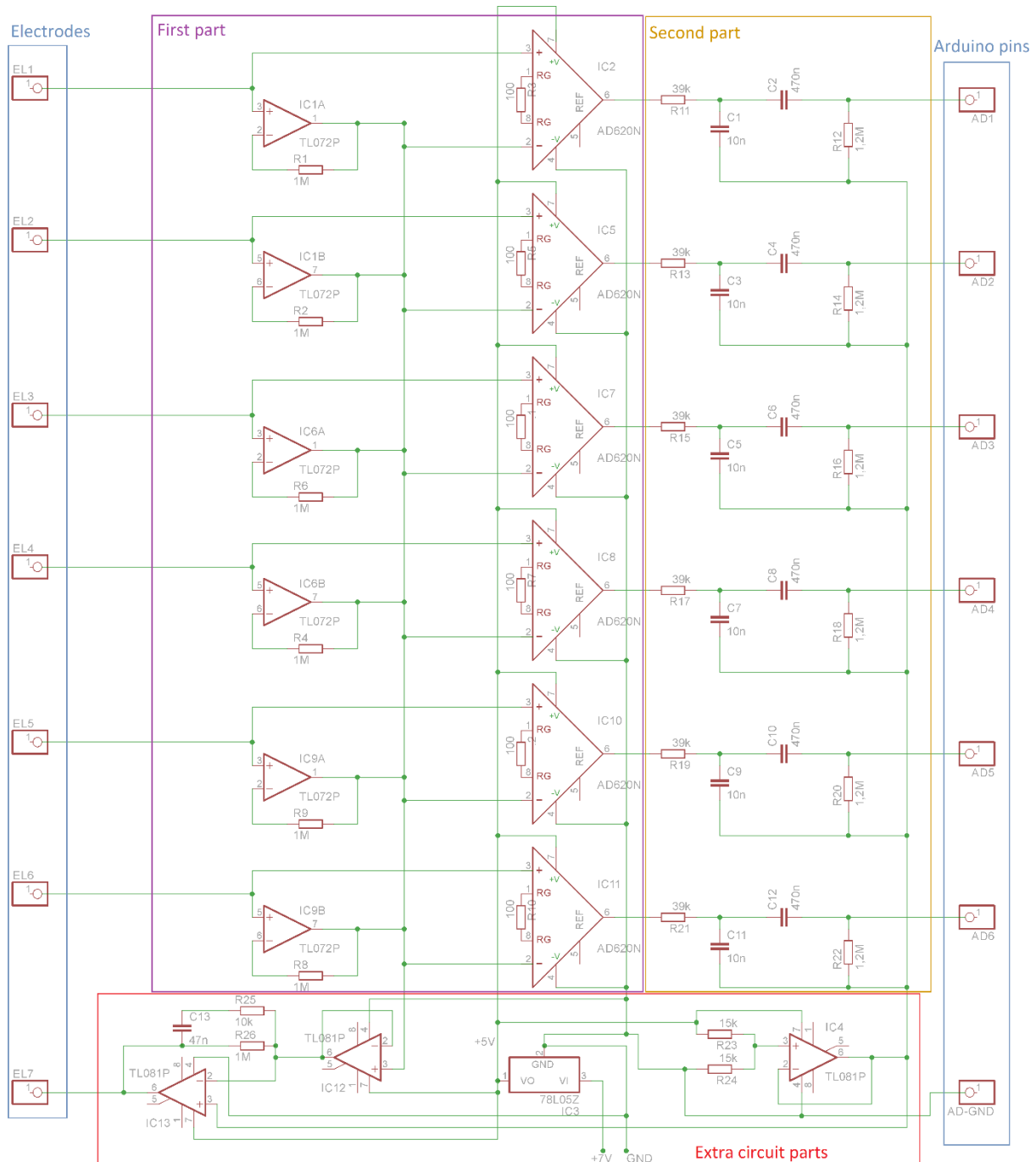


Figure 4: Measuring circuit schematic with highlighted areas

3.1.1.2 First part

The first part, highlighted in cyan in Figure 4, is focusing on the signal leading from the electrodes to the point of real signal creation. From each electrode, two separate pathways lead. First is going into a common centre mentioned before, this also serves to remove common noise in all the electrodes, as main grid interferences and similar occurrences will be cumulatively kept, while all the differences specific for electrodes will be lost. Because of this,

some interferences will be removed or heavily suppressed. This line goes through TL072 operational amplifier, that serves as impedance separator, so following parts will not diminish the voltage of the body as it is a very weak source. Following the amplifiers are 1 M Ω resistors that lower the total amplitude of the signal. This is also used, so the common contains of the signal do not become too strong and cause overflowing of the signal. As mentioned before, several different values of resistors were tested, originally being 10 M Ω , and the best value regarding the results was used. From the resistors, all signals then meet at one node, where they combine into the reference centre point.

A parallel line from the electrodes leads straight into the AD620 amplifiers, where the main measured signal is created.

Following are then parallel instrumental amplifiers AD620, serving both as creators of the raw measured signal, as both the straight line from the electrode and a line from the common centre enter them. They also amplify the measured signal, as ECG has usually amplitudes in microvolts and AD-converters on Arduino have a measuring span of 0-5 V with value step of 4,88 mV, meaning unamplified ECG would be barely registered. These amplifiers were chosen due to their great parameters for biological data measuring even in higher demanded frequencies along with the simplicity of amplification settings. They also serve as impedance separation for the main line from the electrodes. For each measured electrode is one dedicated amplifier serving only one measured signal. These amplifiers are only connected with the common centre reference that enters the inverting input, therefore the resulting signals are not dependant on each other. The amplification was set to roughly 500 using total resistance of 100 Ω .

The resistance setting the amplification was another of heavily changed and tested parts. In the original design, the amplification was even higher, but due to the absence of driven leg and other factors, the outcoming signal from the amplifiers was only rail-to-rail main grid interference with nothing else there. This meant first lowering the signal amplification so at least something would go through undamaged but was later increased after introducing new ways of shedding the interference, such as the driven leg.

All these amplifiers are powered from external power supply, discussed later in Extra circuit parts, setting the power supply to 0-5 V.

3.1.1.3 Second part

Following the amplifiers is a series of analog filters [8], each of the signals having dedicated independent series. In Figure 4 this part is marked with orange. All of these filters are simple first order T-blocks, as heavy filtration is not required, and any precise filters always have better results in digital form. First in the series is a lowpass filter composed of 39 k Ω resistor and 10 nF capacitor, giving a cutoff frequency of 408 Hz. This filter mostly serves to prevent aliasing. A simple first-order filter is used, as the cutoff frequency is lower than half of the sampling one, alleviating the need for steep transduction decline and as using better active filters would also mean the need for amplifiers, that were already high in demand and low in

stock. Also, frequencies this high are not very interesting and important for the main algorithm, so they can be removed.

Second filter in series is a first order highpass T-block composed of 470 nF capacitor and 1,2 M Ω resistor, creating a cutoff frequency of 0,282 Hz. This filter is used to remove baseline drift and some very low frequencies, that could affect the measurement. A simple filter is used again, as the cutoff frequency is very low and most of the undesired noises can be filtered out digitally, if they don't affect the measurement quality. All of the filters are grounded into a virtual ground artificially created only for them described in Extra circuit parts.

3.1.1.4 Extra circuit parts

The supportive parts of the circuit in Figure 4 marked red, through which the main signal doesn't flow, are still very important, as they create the needed conditions that the main part of the circuit requires.

First part is powering the entire circuit. In first iterations, this was done from the Arduino board, as it provides both supply voltage of 5 V and ground. This was later abandoned due to instabilities on the voltage supply, probably caused by the serial communication.

Instead of that, positive voltage regulator was used, in this case L78L05C powered by laboratory generator. The generator serves as reliable power supply, while the L78L05C precisely regulates the voltage flowing to the circuit to 5 V, while also providing stable ground. In addition, the regulator also serves as a rudimentary surge or short circuit protection, as in case of power surge, more than 200 mA would melt it and render it unfunctional, the latter is implemented directly in the part itself. [16]

The whole design of the power supply is also made in preparation for different power source, as the generator can be later replaced by battery or other smaller and portable generator.

Second extra part linked to the power supply is creation of virtual ground for the filters. This need arises from the supply and settings of the amplifiers, as they require positive and negative supply, giving a result around zero. As in this circuit negative voltage is not used or required (also not supported by AD-converters), the supply was done by using the general ground as negative supply, which essentially means that zero on the amplified signal is really equal to 2,5 V against the general ground. And as filters need a ground equal to the signals theoretical zero to operate, one was artificially created. The design is very simple using two resistors and one TL081 amplifier. The two resistors of identical value each come from one branch of the power supply (one from 5 V, second from ground) and meet in the noninverting input of the amplifier. In this point the voltage is averaged and the virtual ground is basically created, as the extra step of amplifier only serves as impedance separation, so the filters don't influence the main supply.

Third part, and also most important one for the signal quality is the driven leg. As mentioned before, this part is used to suppress common interferences in the measured signal right on the body along with removing the base line drift, which in this case was crucial. The source signal for the driven leg comes from the common centre created from the electrodes, which

essentially contains all the major interferences. This signal is then separated (impedance separation) by a TL081 amplifier in voltage observation settings. This signal is then sent to another amplifier with a filter in the feedback loop. In this case, the signal goes to the inverting input, so the result sent to the body is inverse and subtracts from the interference.

3.1.1.5 Data acquisition

In data acquisition, several steps are important. First being reading the values by series of analog-digital converters (ADC) that transcribe the infinitely precise values of analog measuring into digital numbers. Second is then successfully sending this data into the main computer for storing and processing. As these tasks are extremely complicate to create anew, already existing integrated circuit board with dedicated microchip was used, in this case Arduino Uno. This microcontroller has six integrated separated ADCs along with functioning IDE and easy programming based on C. One of the problems that Arduino poses for use in higher prototypes is, aside from the number of analog inputs, the fact that each analog pin can only be accessed independently, meaning that reading all six values need to happen serially, which introduces time slipping errors, as the same events will appear at different stage on each electrode. Thanks to modifying the inner protocols for data reading, this difference is only in magnitudes of microseconds so in a model of this size and quality the difference is negligible. Concerning the operating code and communication with the computer, the program works as following while the entire operating loop run must be done in under 500 μ s. First, outside of the operating void loop, adjustments are made to the protocols of reading the analog data. This was a necessary part, as normally accessing only one value takes approximately 100 μ s, which for six values and serial communication on top is too much. With the modifications applied, reading one value now takes roughly 1-2 μ s. This modification slightly reduces precision, but with the used sampling frequency and precision it's negligible.

Next thing, still outside the void loop, is initialisation of all needed variables, including starting time variable for measuring the time of cycle run, counter for later control of correct data transfer initially set to 0 or a struct containing all the data for sending to the computer. Lastly, the serial communication over USB cable is initialised.

The main operating loop starts by reading the inner time of the device and storing it in the start variable. Followed by that is creating the data struct by reading the analog pin values directly to their positions in the struct with last place being taken by the count number. Then, after transcribing this struct into bytes, all of it is send to the computer with serial function "write". Normally, for serial communication, functions "print" or "println" are used, as they send the data in more complex form, but the Arduino IDE can decode them and immediately show them. However, these functions take longer time to transfer the data (sometimes almost twice) and they cannot send multiple instances or structures like the one used here. Because of that, to send all six measured values these functions would need to be called six times, which would severely overstep the given timeframe. Using the "write" function is then much faster, but also

creates the issue of the transferred data being unannotated bytes, that need to be manually decoded in the computer.

After all the data is transferred, which takes approximately 200-250 μs , the sample counter is increased by one. Then, current inner time is saved into another variable "time" for creating a delay to set the whole loop runtime to exactly 500 μs . Arduino has an implemented delay function that is widely used, however this function, for some reason, never delayed for the precise amount of time, sometimes differing in tens of microseconds. For that reason a special while loop was implemented at the end, where the running condition for it is that the "time" variable is lower or equal to start time variable plus 492 (value chosen by testing as setting it to 500 would cause longer runtime). Inside the loop is only saving the current inner time into the "time" variable. Each iteration of this while loop then creates a delay of around 8-10 μs and runs reliably until the runtime of one entire loop is exactly 500 μs . The entire loop starting with taking the start time is then running perpetually until the serial communication is terminated from the computer, meaning that the length of the measurement is purely dependant on control computer.

3.1.2 Model improvements

The main improvement designed, but unfortunately not fully tested is measuring more data. In this case, by using bigger microcontroller, Arduino Due [17], with twelve dedicated ADCs. This improvement will be used for two different tests. One being higher resolution planar reconstruction. That should yield similar results as the original design, but with better accuracy. The second would be a three-dimensional reconstruction achieved by creating two measured circles on top of each other.

From the second part, an interesting idea arose concerning the placement of the electrodes. The main plan is to place the electrodes of the two rings directly above one another, meaning their position in two planar reconstructions would be the same. The second idea is to place the electrodes of the two circles far apart (with same vertical height), meaning that one electrode would have the same distance to two electrodes of the other ring. This theory will be tested in further development, but not in this thesis.

The main measuring hardware is basically identical, with only duplication of First part and Second part for the new six electrodes, with all else remaining the same.

3.2 Data preprocessing

Digital processing of the measured data has several steps, each of them as important as the other. It all starts with a script creating the communication channel with Arduino and loading the data. That is followed by unpacking and reconstructing the data, as the arriving communication is in unannotated bytes, so it needs to be separated and reconstructed into real number values that can then be processed.

Next step, apart from visual control during the data measurement, is cleaning the data and creating the real bipolar ECG recordings, that are used for the reconstruction. Here arises an interesting question regarding the order of these steps, that unfortunately will not be answered here. The question is, if we have electrode potentials recorded and we need to create the bipolar recordings digitally and also filter them, can the filtration be performed on the potentials alone before creating the whole ECG? As mentioned before, this question remains unanswered here, but maybe during further development a clear answer will be found.

The last step of the preprocessing is creating number vectors, representing the tomographic projections, containing the correct ECG recordings in correct places.

3.2.1 First prototype

3.2.1.1 Serial communication and data reconstruction

For measuring the data, three sequential functions were created. First ensuring the acquisition of the data from Arduino, second translating the bytes into real value vectors and third for plotting all of the measured values to visually ensure, that no error occurred somewhere on the way and the data are not defected.

The first function starts by initialising the serial communication with Arduino with set baudrate of 1000000 and no timeout. The baudrate cannot be changed as a mirroring value is used in the Arduinos internal code. This is followed by a delay period of half a second, as Arduino takes some time to initialise. Then the input buffer is cleared, so no leftover values from previous measuring disturb the incoming data. An empty list is created for storing the loaded data. Then iterating through a “for” loop with number of iterations set to 20000 (resulting in 10 second recordings, can be changed). In each iteration a function “read” is called, set to read 14 values per calling. This function returns a list of 14 bytes incoming from the Arduino, those are then added to the end of the previously created list. The read function without preset delay has an advantage, that when the buffer is empty, it waits for some time until required number of bytes arrive. Once the loop is finished, the serial communication is terminated and the original list, now with 20000 lists inside is returned.

Followed by that is the translation function. Here an empty list for data storing is created, followed by a first “for” loop. In this loop, iterating over the 20000 elements of previously created data, new list as allocated called “values” for storing reconstructed values for each time instance. Followed by that, the current element from measurement is transformed into a list of values thanks to a function that translates each byte into corresponding number. Then a second inner “for” loop follows where in each iteration two neighbouring bytes are taken, always in pairs with first being odd (meaning first and second is taken, then third and fourth etc.). From this pair a single value is created by adding the value of the first byte to a 256-fold of the second byte. This value is then added to the end of the list “values”. When the inner loop ends, the resulting list of values is added to the end of first empty list, ending the first loop. After this, the resulting list of 7 columns and 20000 rows is transformed into numpy array for easier work.

Last comes a simple function, where all the seven columns are plotted, each into separate subplot and shown. In first six subplots, measured data is visible, but mostly not comprehensible due to severe interferences. In the last subplot is the counter value, that looks like a straight line. Thanks to the counter is easily visible, whether any error occurred on byte levels. Then from each electrode can be observed potential data content, as an almost empty reading can also occur, where the values oscillate with difference of one or two.

3.2.1.2 Filtering and real data creation

As mentioned before a question stands, whether it's feasible to filter the pure potential recording and then create the data or vice versa. In this case, due to workflow reasons the ECG was first created, simply by subtracting one lead from another. In total 15 unique recordings are created, usually by iterating over the measured potentials. The way of creating the recordings is basically a digital version of standard ECG creation with amplifiers, taking one potential against other, so just one potential values minus values of the other one.

As for filtration, due to issues with severe interferences, standard linear filters were not able to provide sufficient results, therefore different approach was needed. The best way for this would be a wavelet-based filter, however implementation of one is not as easy, especially for a data this specific and the lack of time. For these reasons, in the first prototype a real filter was not integrated apart from two IIR band-stop filters removing 50 Hz and higher harmonic frequencies. The following approach was purely coincidental and in other scenarios would not work. The main way of getting a signal fit for the reconstruction was a SWT decomposition of the signal to a fifth level, where the coefficients of the deepest level were used as the working signal. In a stroke of luck, in this case these coefficients are extremely similar to a real ECG both in shape and, thanks to use of SWT, length as well. This decomposition along with the linear filtration was done on each of the 15 signals and then send to the core algorithm.

3.2.1.3 Data sorting and vector creation

The following steps are a tight part of the reconstructive algorithm, as they are performed independently for each reconstructed frame. No automated and iterative approach was done yet, so all the values were assigned by hand.

Firstly, the reconstructive algorithm uses six angles, from which the result is reconstructed. For each angle a value vector needs to be created, basically representing the tomographic projection. The correct recordings were by hand assigned for each projection angle and into correct place, meaning that in angel of zero degrees values of 1-2, 6-3 and 5-4 (representation of the ECG, meaning number of the electrodes from which it was created) were used.

The vector for reconstruction itself was made based on the idea, that the measured signal is cumulative and comes from different places and further away the source is, the lower the measured value. Based on that, a zero vector is created and onto it a Gaussian multiplied value of each lead is superposed, centred on preset place. This creates a vector changing

values in which the additions from each lead can influence each other and allowing a dynamic value change during the reconstruction.

This vector has length of 60 samples (resulting reconstruction is 60 by 60 samples), each Gaussian has a specified centre, for some angles three, other angles two, visible in Figure 5 and preset width of 7 equal for all. The width is an experimental value, that can change in future testing.

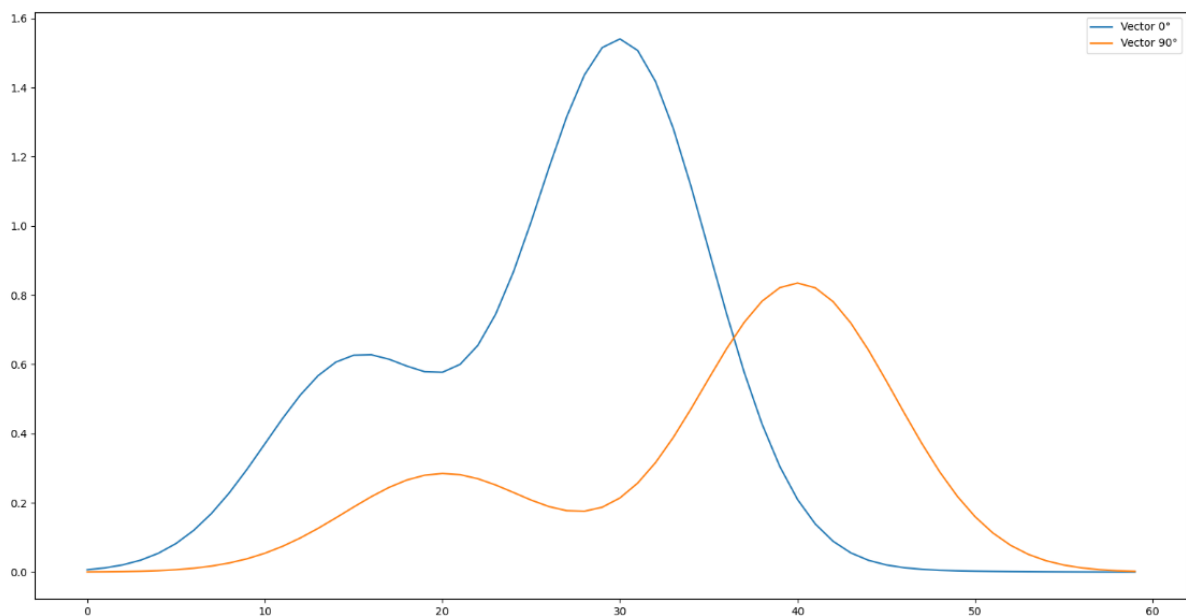


Figure 5: Example of two vectors used for reconstruction

3.2.2 Improvements

3.2.2.1 Filtration

The main improvement done on the original data pre-processing was creation of working filter based on SWT, however the filter itself still has some issues and the results are not always perfect.

The implemented filtering function first removes 50 Hz interferences and its harmonics through a loop using simple butterworth bandstop filters, then with lowpass removes too high frequencies. Then all the signals are padded with zeros to fit the required dimensions of the wavelet decomposition (divisible by 2 to power of the used level). After that, the decomposition is performed using SWT.

Then comes computation of the thresholds. Three different approaches are used, resulting in three different thresholds, which can be chosen in function input. First threshold uses a standard deviation from the whole signal multiplied with a value changeable in function input, by default set to 3,6. Second option is by calculating standard deviation from a specific area. This area needs to be chosen by the user and should be a space between QRS complexes,

as they influence the result a lot. Again, the resulting value is multiplied by the same value as previous. The third option is using standard deviation estimate from median.

The main part is then done in two for loops, one iterating over the decomposition levels, second over the individual samples of each level. Each sample is then tested, if it's smaller than the threshold, its value is set to zero. If it's bigger, it is then changed, as hard thresholding is not implemented. There are two options for handling the used samples, currently used is soft thresholding, but Garrote approach is also present, but needs to be switched on in the code.

Once all that is done, the signal is reconstructed using ISWT, then the padding is removed, resulting in filtered signal the same size as the original.

Several things can be changed in this function by optional inputs, including the level of decomposition by default set to 7, which approach to use for threshold calculation originally the third, the value multiplying the threshold now set to 3,6, in case of second approach the indices of area that should be used, and lastly the used wavelet, currently being sym4.

The resulting filter performs decently, but not perfectly, resulting in some discrepancies in some of the signals, mostly in the QRS complex, but overall is a decent replacement of the originally used approach and useful for proving the algorithm works on real filtered data as well.

3.3 Reconstruction algorithm

The reconstruction algorithm combines several different approaches, most of them modified to fit the needs and basic ideas of it all. In general, it has two main steps, first being a modified version of tomographic reconstructive algorithm for creation of the first model, second is then iterative augmentation of the resulting model according to the main ideas and principles, slowly heading towards the final model.

As this model is creating a time-variant reconstruction, each reconstruction is calculated from a short time fraction of the recordings, these models then put together time series, that visibly shows the electrical activity changes in the chest, mostly on the heart.

The main values used for the reconstruction are differences between two points in each lead. This difference is then used as a single value for each lead creating a projection vector described in 3.2.1.3.

3.3.1 First prototype

As suggested before, this algorithm is performed iteratively, each small time series from the recordings giving out one 2D image. As these images are combined, a video is created.

The first prototype uses for the creation of each model three consecutive samples, creating a difference between the first and the third. The whole reconstruction runs through a "for" cycle, iterating from one to length of the signals with step of three. Two different representations of the final reconstructions were created, as each presents a different view on the data, but also brings certain disadvantages.

First, zero matrixes are created, to allocate the memory and simplify saving the data, with dimensions 60 by 60 by third of the signal length for the first representation, that is visualised with grayscale values, and 60 by 60 by 3 by third of the signal length for the second representation, where a coloured image is created, with each of the colours having a specific movement direction assigned. Third matrix is then created with dimensions of 60 by 60 by 2 by third of the length of the signal, creating space for the vector representation.

3.3.1.1 Reconstruction

In current state, the reconstruction is only performed on a small part of the recorded data, basically one cardiac cycle containing the QRS complex and T wave, as reconstruction of the whole recording would take very long and would not be very useful. For this reason, a part of the signals is first selected with length divisible by three (to prevent indexing issues). The main reconstruction is then done in a loop. In each cycle, a desired part of the signals is selected with length of three samples each. The matrix of these values is then sent into a dedicated function, where all the calculations are done, returning three separate matrixes, one for greyscale, one for colour and one for vector representation. These are then placed into a corresponding place in previously created matrixes.

Inside the main function, the vectors for reconstruction are first created. Firstly, differences between first and last sample (in this case first and third) are calculated to be used as the main values of each recording. Then six vectors are created as described in Data sorting and vector creation. These vectors are then placed into a list, allowing for easier calling during the cycles. Followed by that, two additional lists are created, one containing the numerical degrees of all the planes angles that will be created, as angle affects what is present in the recordings. Second list contains angle values in radians for calculation of "weights" of different projections in each plane. Also a zero matrix is created to serve as a storage for the reconstructed data. The main reconstruction is a sequence of embedded loops, each serving a different purpose. The first loop iterates over the angles of the projections, that will be created, in this case six. In each iteration, a temporary matrix is allocated for the current reconstruction. At the end of this loop, this reconstructed plane is stored into the bigger matrix.

The first embedded loop, iterating from -2 to 2 creates the base first reconstruction. This peculiar numbers serve for better selection of the used vectors from the list. Inside of this loop, the current vector is selected and augmented. As mentioned before, the angle of the recording is important for the values, so a weight system was used, that each vector is multiplied by cosines of angle the current vector holds in the current plane. Also a multiplication of specific factor is in place, ensuring that the used vector has the desired polarity. From this modified projection, a matrix is then created by repeatedly stacking this vector. This matrix is then rotated to the desired angle and added to the base of the reconstructed model. After this loop is done, the first reconstruction is finished, but the results are not very good, hence the reason for further modification.

Following the first loop is a presetting of values for the next loop, such as iteration counter, initial error rates, step for modifications and new vectors with gaussian curves, this time with width of 30. Again, these five vectors have the same centres as the original ones, but the vectors are reconstructed in the following loop.

Now starts the augmenting loop, this time a while loop with conditions of error rate being bigger than 0,3 and number of iterations lower than 200. In the start of this loop, error rate is set to zero, as it's continually added during computation. Also a zero matrix is created for reconstruction of the error values. Then follows an internal for loop, similar to the previous one for reconstruction. In here, back projection is created from the first reconstructed image, by calculating a sum over one axis when the original image is turned to a desired angle. Also the original projection corresponding to this angle is loaded. Then follows a condition, deciding if the original projection has two or three original values. Inside both possible conditions, similar process happens, where for each centre-point a corresponding gaussian multiplied original projection is calculated, then its sum. Then a ratio of sum of both original and newly created projection is calculated along with difference between these two vectors. The final vector is then created by multiplying the ratio, difference and corresponding gaussian. From the resulting two or three vectors the whole error vector is created by superposing them on one another. The sum of absolute values of this vector is then added to the error counter.

From this error vector, a matrix is created and rotated to desired angle, just like in the original reconstruction, with additional zeroing of pixels, that already have negative value in the reconstruction. This prevents creation of unwanted negative spots that are compensating for higher values around (This needs to be changed in the future, as it negatively influences reconstructions with intentional negative values and is not very accurate). Lastly the resulting error matrix is added to the reconstruction of error values.

Once the for loop ends, the original reconstruction is saved into another variable, in case that the next iteration worsens the results. Then to the original image the matrix of error values is added, multiplied by the modification step. Lastly, a convergency safeguard in form of condition is placed, that stops the modification loop if the error rate increases, while returning the previous reconstruction with the better error rate. After this, if cycle continues, the current error rate is saved into a new variable to be used in next iteration. The whole loop is finishing with iteration counter being increased by one.

When the while loop ends, the resulting image is then saved into its specific place for the reconstructions, also finishing the very first for loop.

Once all the reconstructions are calculated, the final representations are created from them. The image for greyscale is made by calculating a sum of each pixel over the six planes, giving a simple 60 by 60 matrix. This is very simple, but poses an issue where certain structures may not appear due to opposing values in different planes (which can correctly happen).

Due to this issue, values for coloured visualisation are also calculated. This system also poses a problem, as the main colours are three, but the reconstruction only has two main axes. Because of this, non-orthogonal colour system is used, where each colour vector bares angle of 60 degrees with the other two. The values for each are created by adding three

reconstructed planes together, the main one with the corresponding angle and two neighbouring ones.

Lastly, the vector representation is created where two matrices of values, one of x axis movement magnitude, the other for y axis, are joined together. Each of these matrices is calculated by adding all the reconstructed planes multiplied by cosine of angle of the plane. This ensures the main idea of angle dependency, while removing the influence of the perpendicular plane. All the created representations are then returned by the function.

Once the results of single reconstruction are returned by the function, they are stored in designated places for further work.

3.3.2 Improvements

The main improvement done on the reconstruction algorithm was introducing a premade mask, visible in Figure 6, onto which the reconstruction should happen. This mask is a circle placed into a middle of maxima in previously done reconstructions with radius of seven pixels. The main line of the circle is also blurred into the surrounding, creating a more gradual increase of values. The highest values of the circles main line are 1,87, while all the background is 0,955. This mask is then used in creating the first reconstruction before modifications are done, where it is multiplied with the used projection matrix. Thanks to the higher values of the circle, the further modification does not diminish them as much as the surrounding, or even the centre, resulting in clearly visible circle in the reconstruction. The surrounding is non-zero so that all the values are preserved, but also not equal to one for purposes of preserving the energy of the image.

The idea of the mask comes from the fact that basically all of the measured and used activity comes from the heart, that has, at least for simplification, circular shape and has no activity in the centre. Different positions of the circle were also tested, resulting in almost blank frames or very defected results.

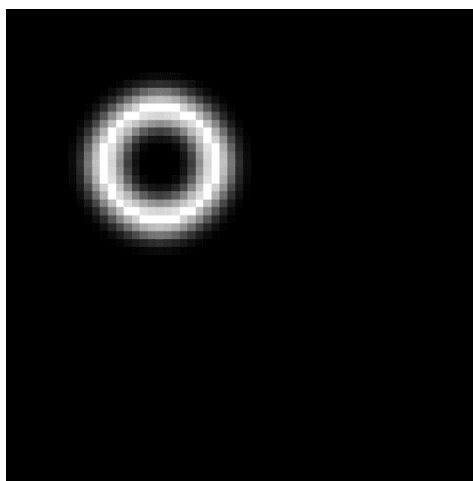


Figure 6: Mask used for improved reconstruction.

3.4 Visualisation of results

Once the reconstruction is done, the resulting data needs to be visualised in appropriate manner. For this, three separate functions were created, each creating animated plot of one certain type (greyscale, RGB, vector). For all of these functions the animation toolbox from matplotlib was used.

3.4.1 Greyscale visualisation

This function is responsible for visualising the greyscale representation in an animation, allowing for the visibility of dynamic changes. Along the animated image, a plot of the used ECG data from single lead is displayed, also dynamically changing for better context of what is happening and what part was reconstructed.

This function takes two mandatory inputs, the reconstructed data and original leads for visualisation, and two optional inputs used for saving the created animation, one being boolean value whether to save, second a name under which to save the file.

As mentioned, the process for creating the animation uses the matplotlib animation toolkit, in this case the “ArtistAnimation” function. This function creates the animation from already pre-created series of plots. Because of this, after presetting necessary parameters for the figure, a loop is placed, that iteratively creates the plots and saves them into a list, that the animation function takes as input argument.

The animation itself has set time between frames to 50 ms, resulting in frame rate of 20 Hz. Also a delay between restart is placed, lasting for one second.

The last thing in the function is conditional saving of the created animation. In this case, the data is stored into Gif format. Unfortunately, this format does not support changing its playback framerate, due to that, the played Gif is playing much faster, as it has higher default framerate. Also, the saved Gif appears to have much lower bit depth in terms of colour, meaning that sudden different value steps of larger magnitude appear, that were not present in the program created animation.

3.4.2 RGB visualisation

For the RGB coloured visualisation, basically identical approach was used as for Greyscale visualisation, this time with additional choice of the displayed lead. This is based on optional input “plot_number”, that is set to True by default. When the default value is used, all of the 15 plots are displayed at once. This happens due to a condition in the beginning, where if “plot_number” is true, six subplots are created, one with the reconstruction, the other five with three signals displayed each. The process itself is very similar to the greyscale version, only with several subplots added. The second branch of the condition, activated when number of desired recording is inputted, is basically identical to the greyscale version, only with changed indexing of displayed lead. Last comes a condition for saving the created animation, again identical, and common to both branches.

3.4.3 Vector visualisation

This function, using same inputs as the previous ones, uses slightly different approach. Firstly, the data needs to be prepared, starting by separating the input data into x and y matrices. Then a base for coordinates system is made, meaning two vectors of all values on each axis. This is one of the mandatory inputs in the used function.

Then comes a more complex subplots presetting, as in this case, the vector reconstruction is handled as a graph instead of an image, meaning the result would be stretched by the plotting function (one side much longer than the other). This means that positions of both the plots and their dimensions is set manually. Then follows the initial plotting of both data, where for the reconstruction quiver function is used.

In this case, "FuncAnimation" is used. This creates the animation with a function, that needs to be created and inputted. This function is made so that in each calling, it generates new plots for the current frame. This function is then iteratively called by the animator, creating the changes.

The challenges faced in this function were setting of proper scaling of the quivers, as they can then be too long and chaotic, and also attempts to remove the background values, that can also be disturbing in the visualisation. Unfortunately, both come at a cost, for scaling the issue of small changes not being visible (T wave), for filtering the problem of value inconsistency, causing the final instability of the displayed reconstruction, resulting in it not being used.

Lastly also comes the same condition as in previous functions, allowing for saving the data.

4 Results

When it comes to results, many different things could be listed. For simplification, the sub results of all designed parts will not be listed, only the final reconstructions, but in different levels of development, including first “filtering” approaches, or both methods of reconstruction. In all of the cases, preselected sections of measured signals will be used, always with length of 900 samples or 0,45 seconds. All of the shown reconstructions contain hypertext links for online stored full reconstruction in form of Gif. A thing also worth mentioning is that Gif format diminishes the quality of the resulting pictures, mostly in terms of bit depth. This causes the stored data to be more difficult to see and also creates new artefacts, that are not present in the original visualisation.

Starting with using the original signal preprocessing and first reconstruction model, in Figure 7, Figure 8 and Figure 9 are plotted selected frames from one of the measured datasets. One visible thing is the effect of the wavelet decomposition on the signal, specifically the small ripples and waves before the QRS complex and also its prolonged duration. For the reconstruction itself, it shows the predicted, meaning maxima located in the frontal left side, which is in general the exaggerated placement of the heart. In greyscale (Figure 7) and RGB (Figure 8), the maxima are relatively well visible, in the vector graph (Figure 9), they are also visible, but due to noise in the background, are more difficult to see. Unfortunately, in neither the greyscale nor the RGB are visible the changes during the T wave, most likely due to very low values. Some changes are visible in the vector representation, but one needs to know what to look for.

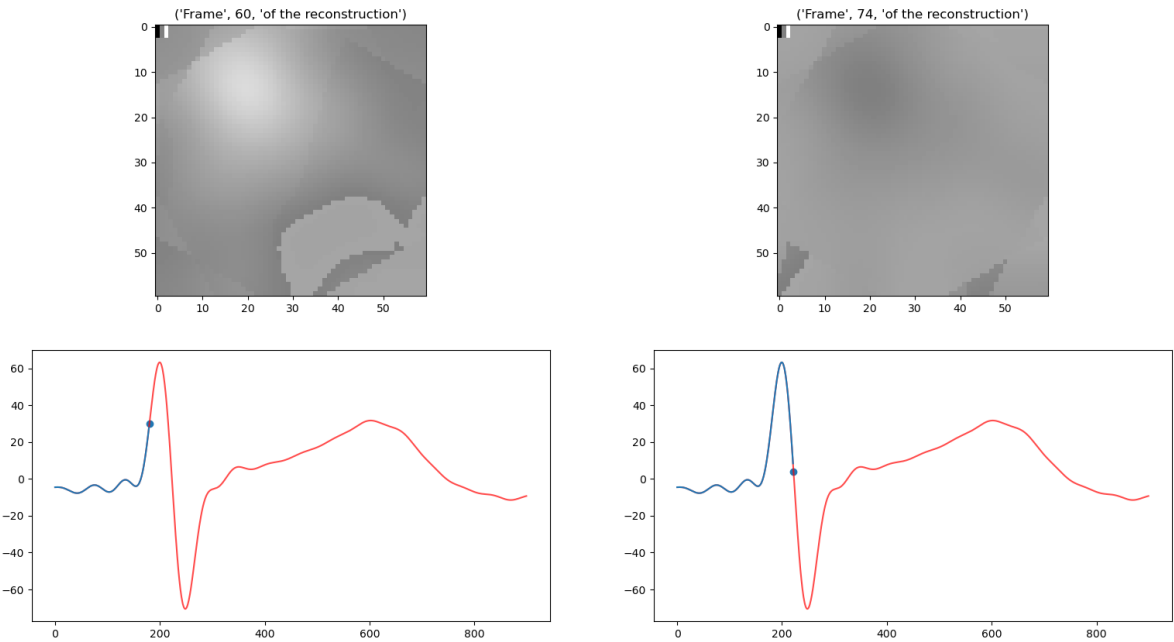


Figure 7: Example of greyscale representation of original reconstruction with original signal processing

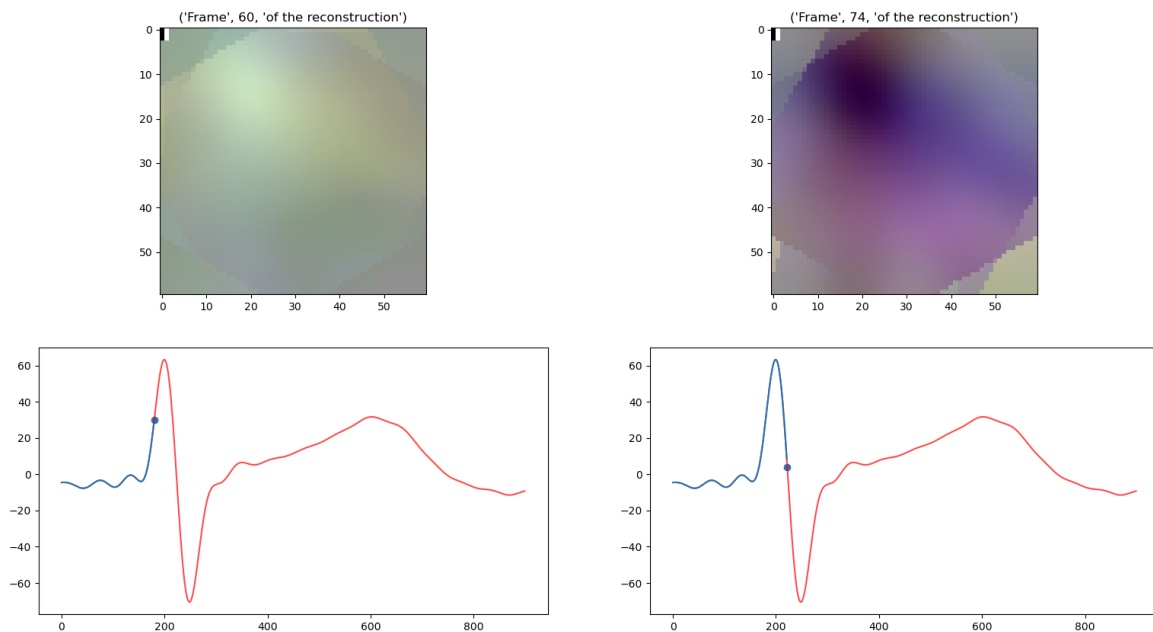


Figure 8: Example of RGB representation of original reconstruction with original signal processing

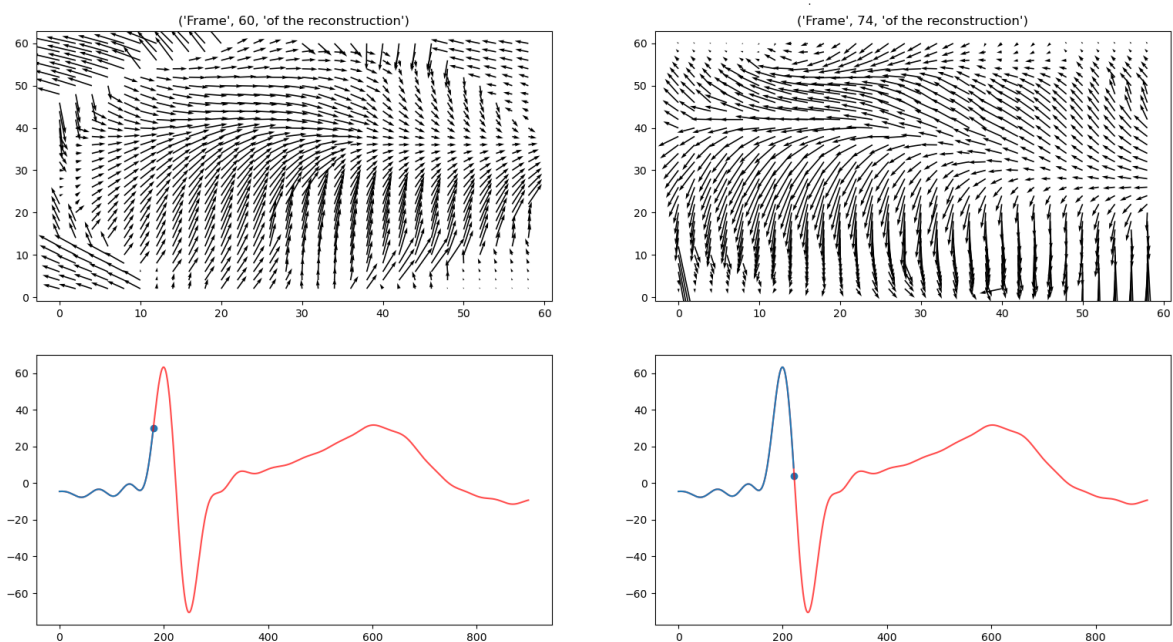


Figure 9: Example of vector representation of original reconstruction with original signal processing

When combining the created real filter with the original reconstructive algorithm, we get results visible in Figure 10, Figure 11 and Figure 12. Here, a better difference can be seen in the shape of the real ECG, with shorter QRS complexes, straighter lines, but also with occasional remaining interferences. However, in RGB (Figure 11) are also visible the shortcomings of the

real filter in form of defective parts in QRS complexes that then result in problematic or incorrect values in the reconstruction itself. Another issue caused by the real filter is the duration of the section of interest caused by short QRS complexes, resulting in very few frames with interesting values. In greyscale (Figure 10) the issue of nonvisible maxima is present in the descending part, previously explained in Reconstruction. In the vector graph (Figure 12), the maxima are again poorly visible due to the surrounding values.

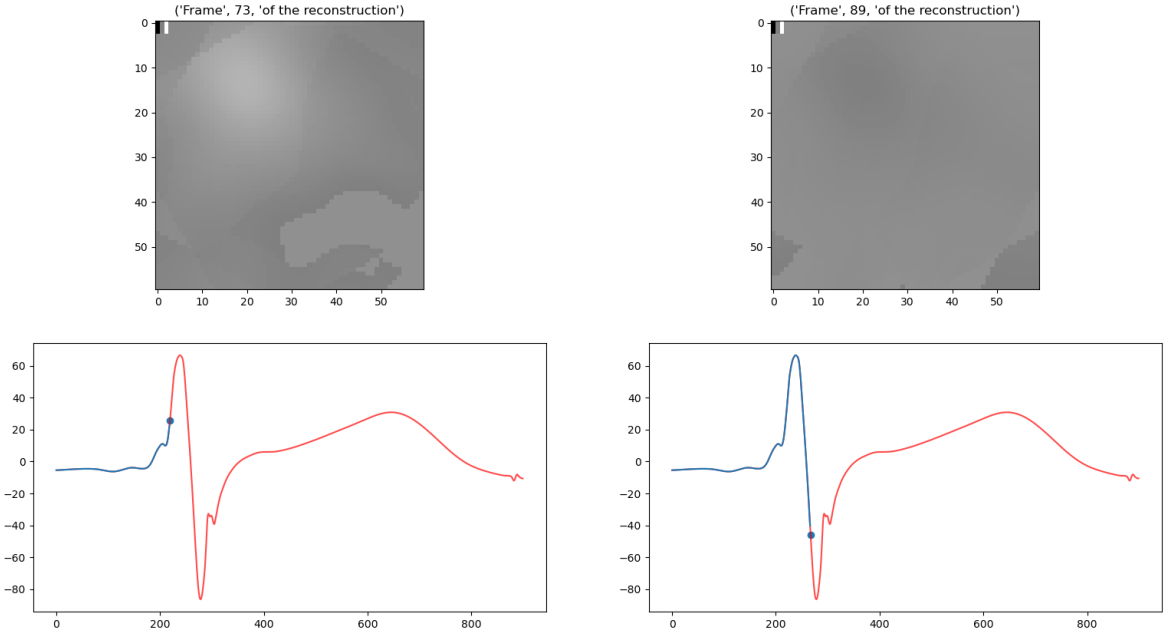


Figure 10: Example of greyscale representation of original reconstruction with filtered signal

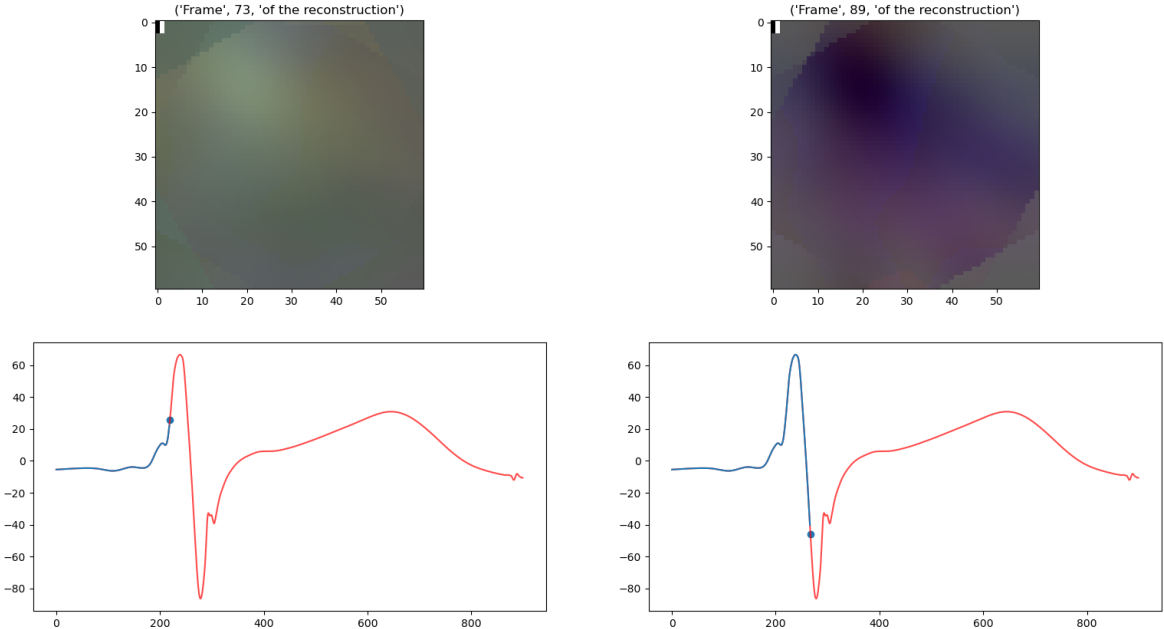


Figure 11: Example of RGB representation of original reconstruction with filtered signal

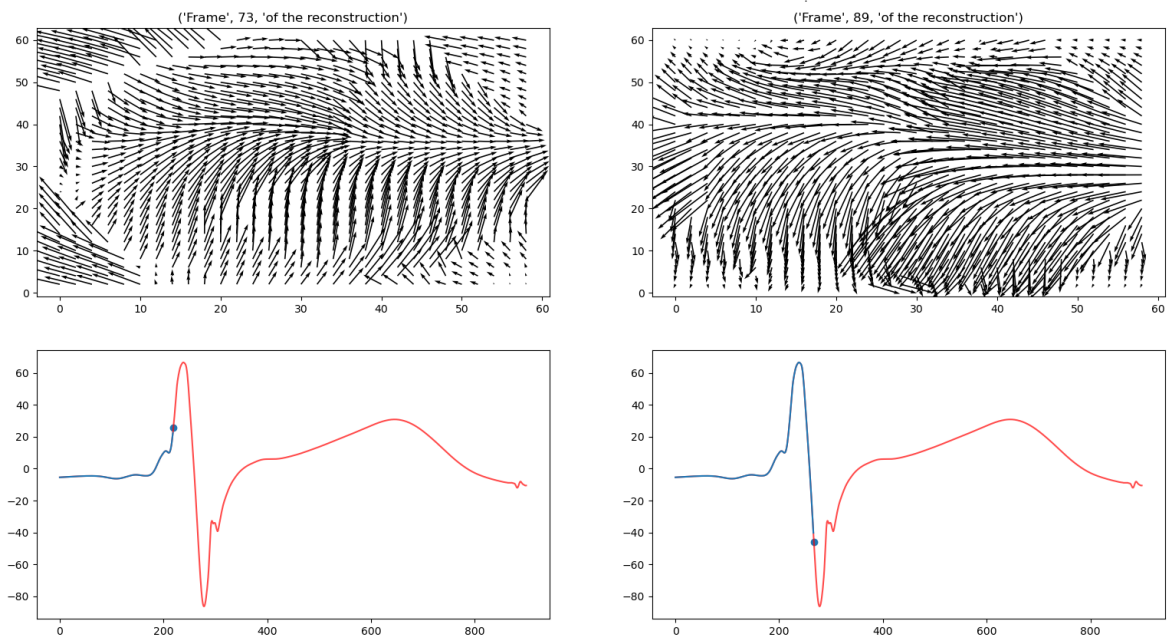


Figure 12: Example of vector representation of original reconstruction with filtered signal

When combining the real filtered data with the improved reconstruction, we get Figure 13, Figure 14 and Figure 15. Here is visible the impact made by using the mask for reconstruction, as the maxima are very well visible. In RGB (Figure 14) the values of ascending part are a bit dimmer, but still well distinguishable from the surroundings. In greyscale (Figure 13) the descending values are again less visible, but unlike the original model, they are visible more. In vector graph (Figure 15) the surrounding values again pose a challenge for visibility, but the ring structure and its activity is better distinguishable. In all three of the recordings, a few faulty frames also appear, most likely due to imperfections in the filtered signals.

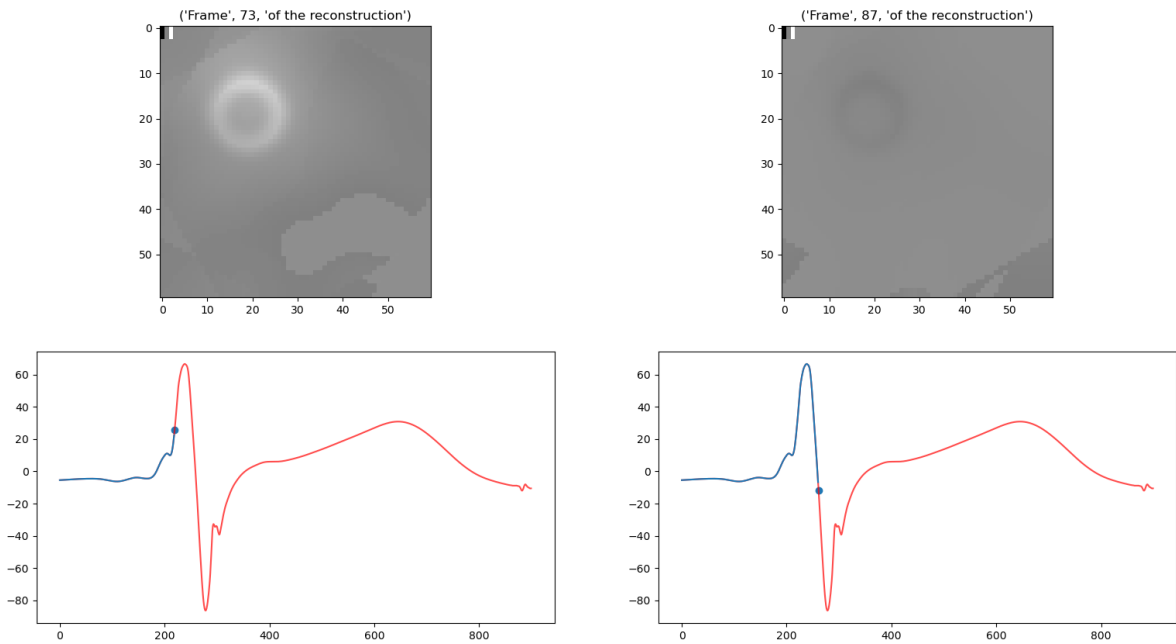


Figure 13: Example of greyscale representation of improved reconstruction with filtered signal

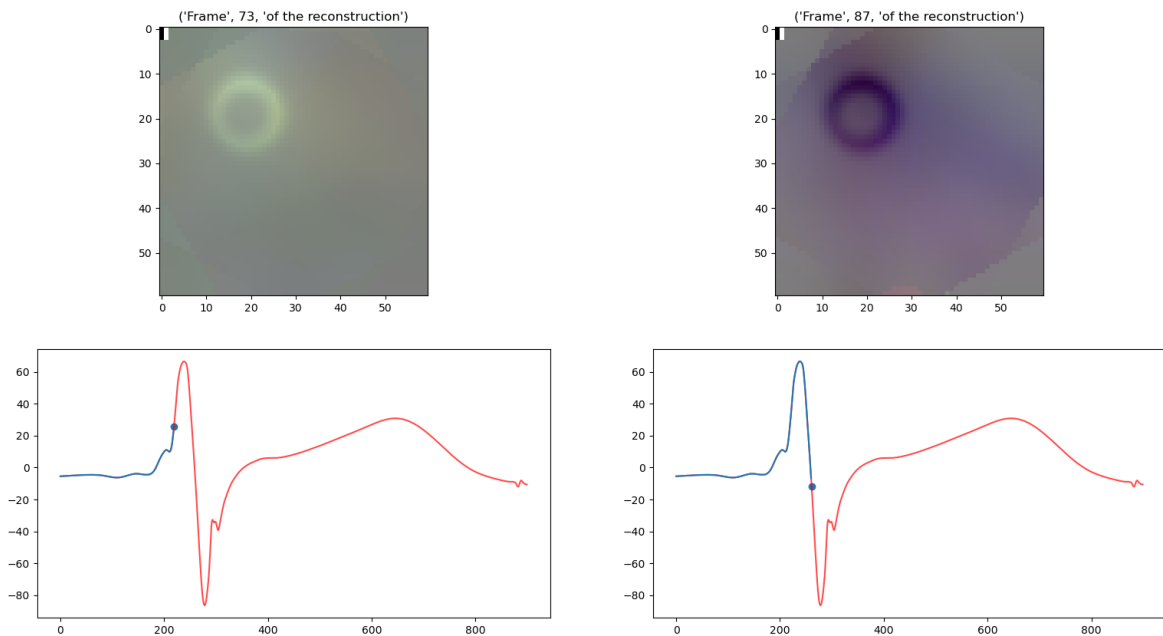


Figure 14: Example of RGB representation of improved reconstruction with filtered signal

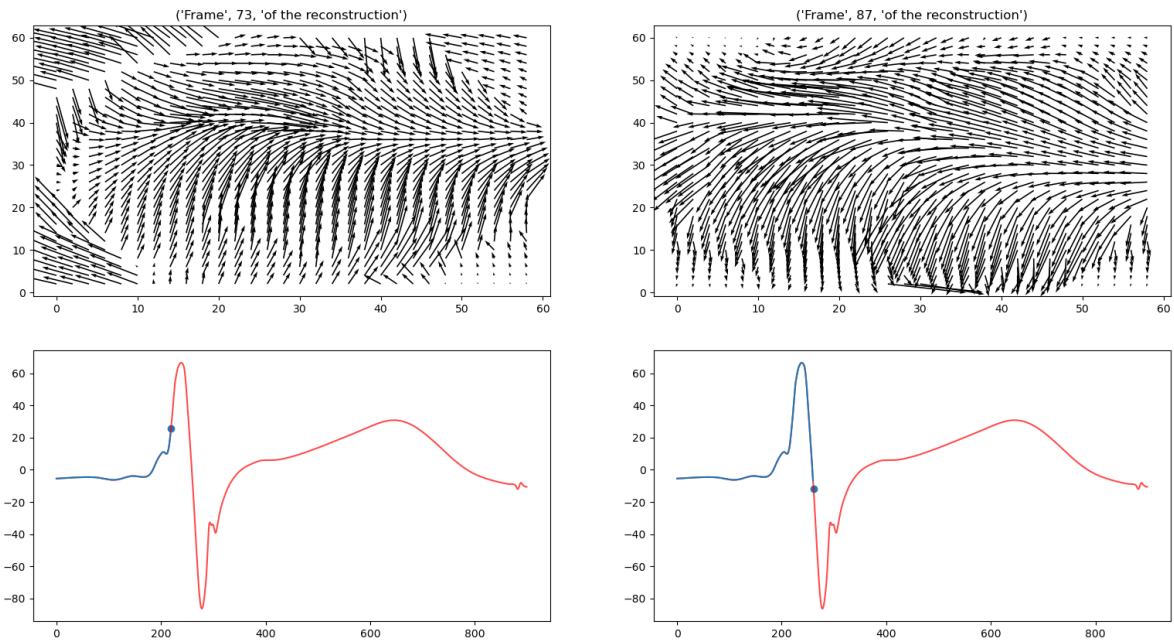


Figure 15: Example of vector representation of improved reconstruction with filtered signal

The best results are then achieved by using the original wavelet decomposition with use of improved reconstruction, visible in Figure 16, Figure 17 and Figure 18. Similarly as with previous, the positive values in RGB (Figure 17) and negative values in greyscale (Figure 16) have a lower intensity, but are well visible. Unlike with use of real filtration, almost no faulty frames are present, as the signal is much smoother. The visibility in vector graph (Figure 18) is again rather poor, but the main structure is still visible.

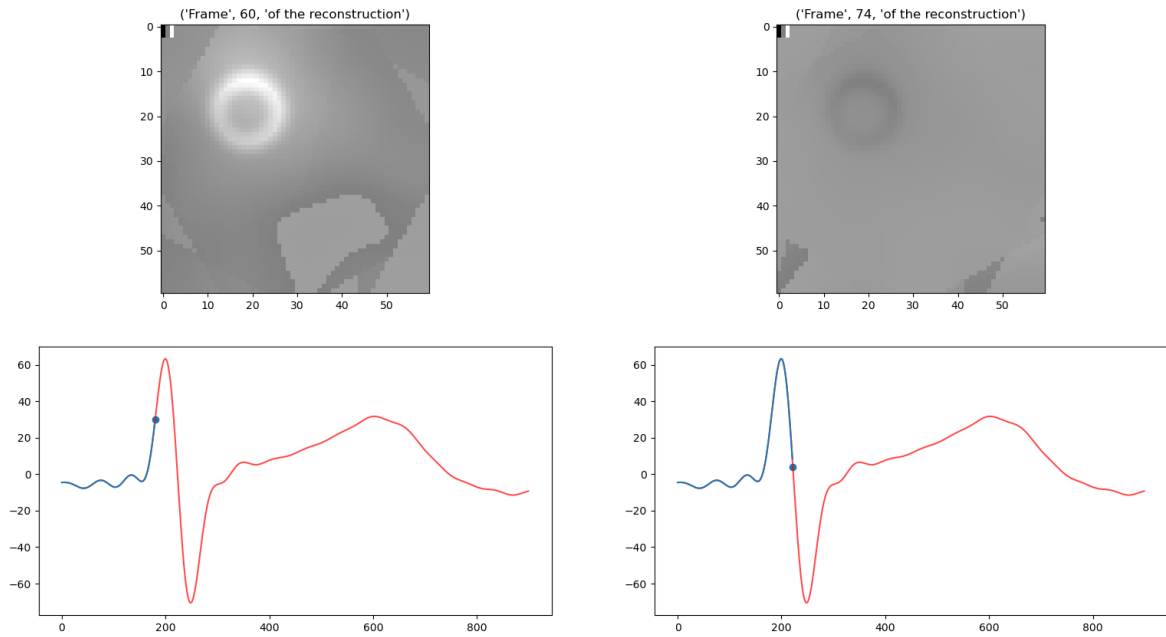


Figure 16: Example of greyscale representation of improved reconstruction with original signal processing

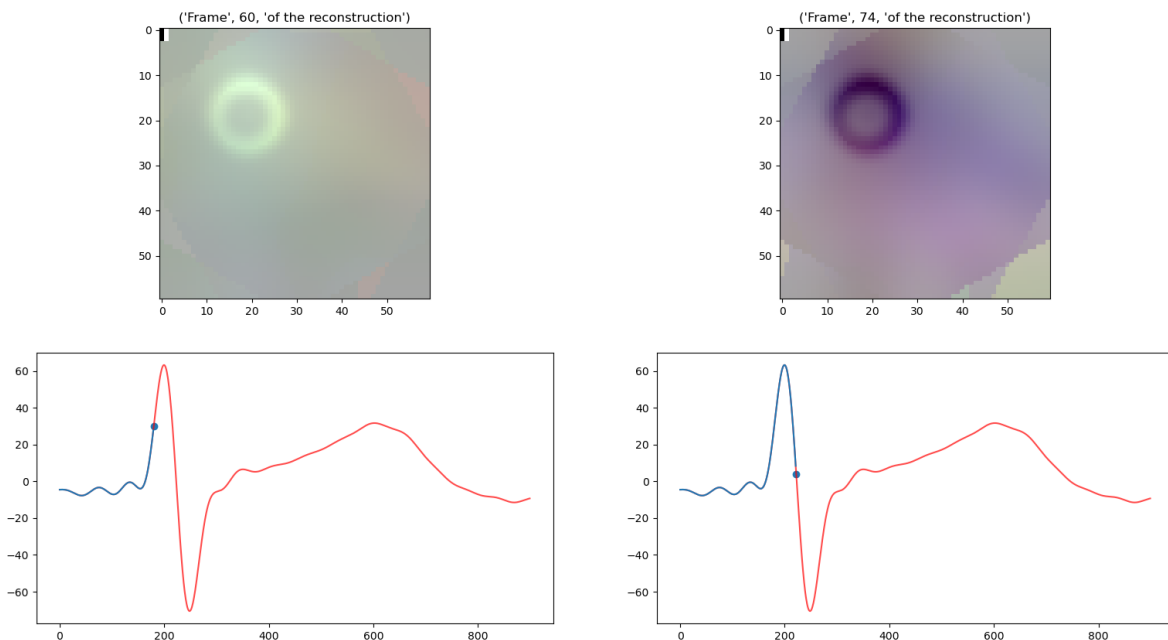


Figure 17: Example of RGB representation of improved reconstruction with original signal processing

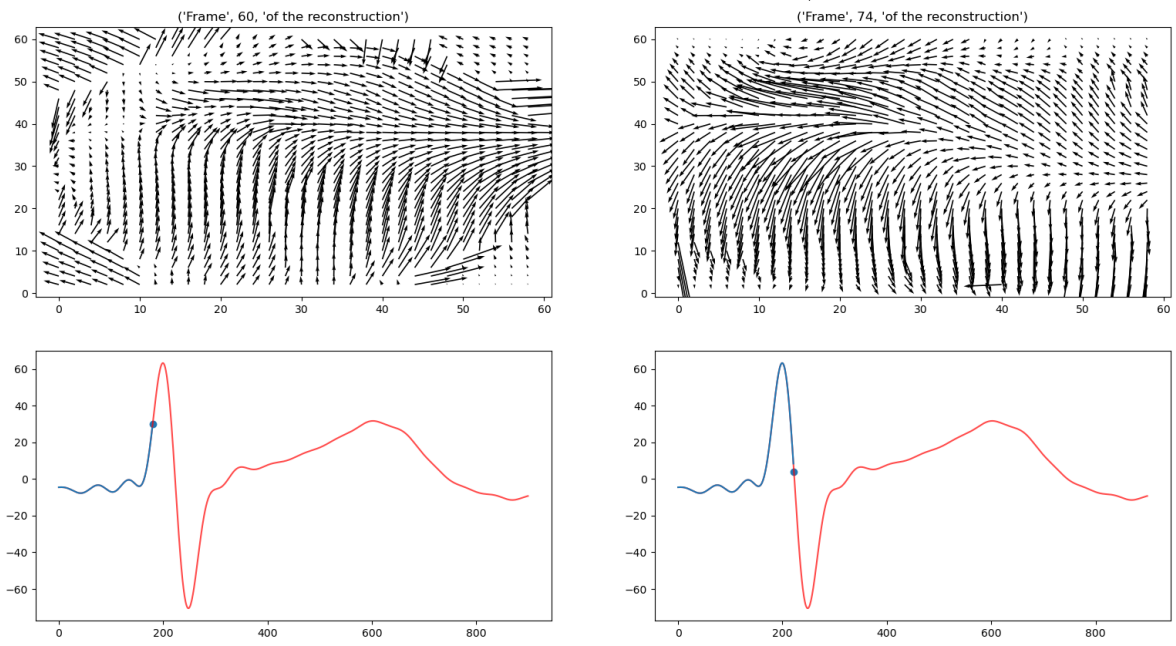


Figure 18: Example of vector representation of improved reconstruction with origina signal processing

5 Discussion

To summarise everything, most of the set goals were achieved. First, the measuring hardware was created, even though with lower quality, still sufficing to the needs of this work. The results of this system are sometimes problematic, as they may create errors, but in repeated measurement provided sufficient data. Unfortunately, due to shortage of some electrical parts and high time demands during development, the 3D measurements were not finalised.

In data pre-processing, creating a specific filtration was achieved, functioning well enough to allow the work to proceed. The general results of the filtration are however unsatisfactory, as first approach doesn't really filter the data and cannot be relied upon in future, the second approach having many issues with correct results, meaning that severe modifications would be necessary in the future.

Concerning the data sorting and creation, an approach sufficient for this stage of development was implemented, however for future work may not be feasible. Data sorting itself was done by hand, at this point the simplest option but for future not possible. Other reason for not creating automated system was issue of its usability in next prototypes, as in higher dimensions it would probably work differently.

When it comes to the main reconstruction, despite many setbacks, the final results are decent. Of course, the depth of development and quality needs to be taken into account. Originally, there was no plan to use a mask for reconstruction, but considering how the results improved, it was implemented in the end.

The means of visualisation are the real issue. Three different were created, each with its strong and weak sides, the final results, however, are not that impressive. They serve their purpose but could be much better (mostly in terms of displaying quality).

As this work serves as proof of concept for future possible development, the current results are not that impressive. However, considering the timeframe, possibilities in terms of hardware and demands in things created anew, the resulting reconstructions so far proved the possible viability of this design. As shown in results, the algorithm can successfully localize the rough centre of the signal origin, and when fitted with pre-created mask, even allocate these places onto a much smaller inhomogeneous area. These results then support a deeper development, that will hopefully result in more precise device.

Many errors and shortcomings are present in the current design, most of them then having part in worsening the results. Some of them are simpler to improve, such as number of measuring electrodes, others are more difficult, like designing a fail-proof filtration system.

Starting with hardware, as mentioned before, more electrodes will help with the possible improvement of spatial resolution. As mentioned in Model improvements, twelve leads will be tested, which still won't be enough. A final aim in the development would be creating a large electrode field across the chest with more than 200 electrodes. Another improvement in hardware, apart from higher quality components, would be parallel sample acquisition, meaning that all the samples would be measured at the same time, negating the effect of time

slipping. Along with this, different methods for measuring the electrodes potentials can be tested, as well as their different styles of placement as mentioned in Model improvements.

When it comes to pre-processing, several things will need to be improved. Apart the data sorting that should be automated, the biggest improvement has to be in data filtration. The currently used system is effective, but still very problematic, creating many errors. A possible way to fix this would be implementation of specialised filter, likely based on tree branch decomposition, that would have a variable threshold and would operate differently in areas of QRS to mostly preserve its shape. Then testing will need to be done to answer a question from Data preprocessing, to see whether filtration of base potentials is feasible.

Another possibility would be to test new methods of data creation. Now, for its simplicity, Gaussian is used to introduce the source uncertainty, however different distribution curves should be tested, such as Students T-distribution or some specially designed curve, to test for possible better results.

Many different improvements will need to be done in the reconstruction algorithm. The main one would be its conversion into a higher dimension, meaning that each reconstructed frame would be a 3D model. This will most likely demand a slight change in approach to the reconstruction, as planes from different rings will also be influencing each other. Another, rather simple change will be lowering the number of samples used for each frame, resulting in better time resolution.

Next possible change is to use an estimation of the measured chest, meaning calculating the reconstruction directly onto an oval or a circle, rather than a square. This would erase the corner artifacts present in current reconstruction and could also improve the quality and accuracy of the final results, as it would more resemble the real world.

Another possibility for improvement would be to create some sort of potential tracking, that would essentially compare neighbouring frames and ensure, that the reconstructed signals would not suddenly appear or disappear and would follow their reconstructed directions.

Last and most important thing would be to perfect the visualisation methods. The most viable system is vector-based imaging, as it can perfectly showcase all the important information and would be best for 3D visualisation. For this to happen, two things will be essential, one being removal of outlying values, so the main areas would be better visible, second would be correct setting of scales.

Bibliography

- [1] J. Malmivuo and R. Plonsey, *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*. 2012. doi: 10.1093/acprof:oso/9780195058239.001.0001.
- [2] “Low cost low power instrumentation amplifier AD620.” 2003. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD620.pdf>
- [3] “The ECG leads: Electrodes, limb leads, chest (precordial) leads and the 12-Lead ECG,” *Ecgwaves.com*. 2014. [Online]. Available: <https://ecgwaves.com/topic/ekg-ecg-leads-electrodes-systems-limb-chest-precordial/>
- [4] M. Borlich and P. Sommer, “Cardiac Mapping Systems,” *Card Electrophysiol Clin*, vol. 11, no. 3, pp. 449–458, 2019, doi: 10.1016/j.ccep.2019.05.006.
- [5] S. Coveney, C. Cantwell, and C. Roney, “Atrial conduction velocity mapping,” *Med Biol Eng Comput*, vol. 60, no. 9, pp. 2463–2478, 2022, doi: 10.1007/s11517-022-02621-0.
- [6] A. Albulbul, “Evaluating Major Electrode Types for Idle Biological Signal Measurements for Modern Medical Technology,” *Bioengineering*, vol. 3, no. 3, p. 10, 2016, doi: 10.3390/bioengineering3030020.
- [7] J. Punčochář, *Operační zesilovače v elektronice*, 5. vyd. Praha: BEN - technická literatura, 2002.
- [8] T. P. Zieliński, “Analog Filters,” in *Starting Digital Signal Processing in Telecommunication Engineering*, Cham: Springer International Publishing, 2021, pp. 155–191. doi: 10.1007/978-3-030-49256-4_7.
- [9] “What is a Zener Diode,” *TTI*. 2003. [Online]. Available: <https://www.tti.com/content/ttiinc/en/resources/blog/what-is-a-zener-diode.html>
- [10] J. Jan, *Digital signal filtering, analysis and restoration*, 1st ed. London: The Institution of Electrical Engineers, 2000.
- [11] L. Smital, “Vlnková transformace elektrokardiogramů,” Dissertation thesis, Vysoké učení technické v Brně, Brno, 2013.
- [12] L. Smital, M. Vítek, J. Kozumplík, and I. Provazník, “Adaptive Wavelet Wiener Filtering of ECG Signals,” *IEEE Trans Biomed Eng*, vol. 60, no. 2, pp. 437–445, 2013, doi: 10.1109/TBME.2012.2228482.
- [13] C. Pravin and V. Ojha, “A Novel ECG Signal Denoising Filter Selection Algorithm Based on Conventional Neural Networks,” in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2020, pp. 1094–1100. doi: 10.1109/ICMLA51294.2020.00176.
- [14] J. Radon, “On the determination of functions from their integral values along certain manifolds,” *IEEE Trans Med Imaging*, vol. 5, no. 4, pp. 170–176, 1986, doi: 10.1109/TMI.1986.4307775.
- [15] M. Pham, Y. Yuan, A. Rana, S. Osher, and J. Miao, “Accurate real space iterative reconstruction (RESIRE) algorithm for tomography,” *Sci Rep*, vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-31124-7.

- [16] "Positive voltage regulators." 2024. [Online]. Available: <https://www.st.com/en/power-management/178l.html#overview>
- [17] "Arduino Due." 2024. [Online]. Available: <https://store.arduino.cc/collections/boards-modules/products/arduino-due>

List of Figures

- Figure 1: Example of simple ECG measuring schematic from AD620 datasheet [2] 8
- Figure 2: Stationary wavelet decomposition with thresholds for each level..... 15
- Figure 3: Example of picture decomposed to sinogram 16
- Figure 4: Measuring circuit schematic with highlighted areas 20
- Figure 5: Example of two vectors used for reconstruction 27
- Figure 6: Mask used for improved reconstruction. 31
- Figure 7: Example of greyscale representation of original reconstruction with original signal processing 34
- Figure 8: Example of RGB representation of original reconstruction with original signal processing 35
- Figure 9: Example of vector representation of original reconstruction with original signal processing 35
- Figure 10: Example of greyscale representation of original reconstruction with filtered signal 36
- Figure 11: Example of RGB representation of original reconstruction with filtered signal 36
- Figure 12: Example of vector representation of original reconstruction with filtered signal ... 37
- Figure 13: Example of greyscale representation of improved reconstruction with filtered signal 38
- Figure 14: Example of RGB representation of improved reconstruction with filtered signal .. 38
- Figure 15: Example of vector representation of improved reconstruction with filtered signal 39
- Figure 16: Example of greyscale representation of improved reconstruction with original signal processing 40
- Figure 17: Example of RGB representation of improved reconstruction with original signal processing 40
- Figure 18: Example of vector representation of improved reconstruction with original signal processing 41

List of Abbreviations

ECG	Electrocardiogram
2D	Two dimensional
3D	Three dimensional
CT	Computed tomography
MRi	Magnetic resonance imaging
Hz	Hertz
HRV	Heart rate variability
ICU	Intensive care unit
CMRR	Common mode rejection ratio
IIR	Infinite impulse response
FIR	Finite impulse response
DTWT	Discrete time wavelet transformation
SWT	Stationary wavelet transformation
ISWT	Inverse stationary wavelet transformation
AI	Artificial intelligence
1D	One dimensional
V	Volts
MΩ	Mega ohms
mV	Milli volt
kΩ	Kilo ohm
nF	Nano farad
mA	Mili ampere
C	Programming language
ADC	Analog-digital converter
IDE	Integrated development environment
μs	Microsecond
ms	Millisecond
RGB	Red-green-blue

A: Code for Arduino

```
#define FASTADC 1

// defines for setting and clearing register bits
#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

struct data{
  int value1;
  int value2;
  int value3;
  int value4;
  int value5;
  int value6;
  int counter;
};
int start ;
int time;
int count = 0;
int starter;

void setup() {

#if FASTADC
  // set prescale to 16
  sbi(ADCSRA,ADPS2) ;
  cbi(ADCSRA,ADPS1) ;
  cbi(ADCSRA,ADPS0) ;
#endif

Serial.begin(1000000) ;
}

void loop() {
start = micros();

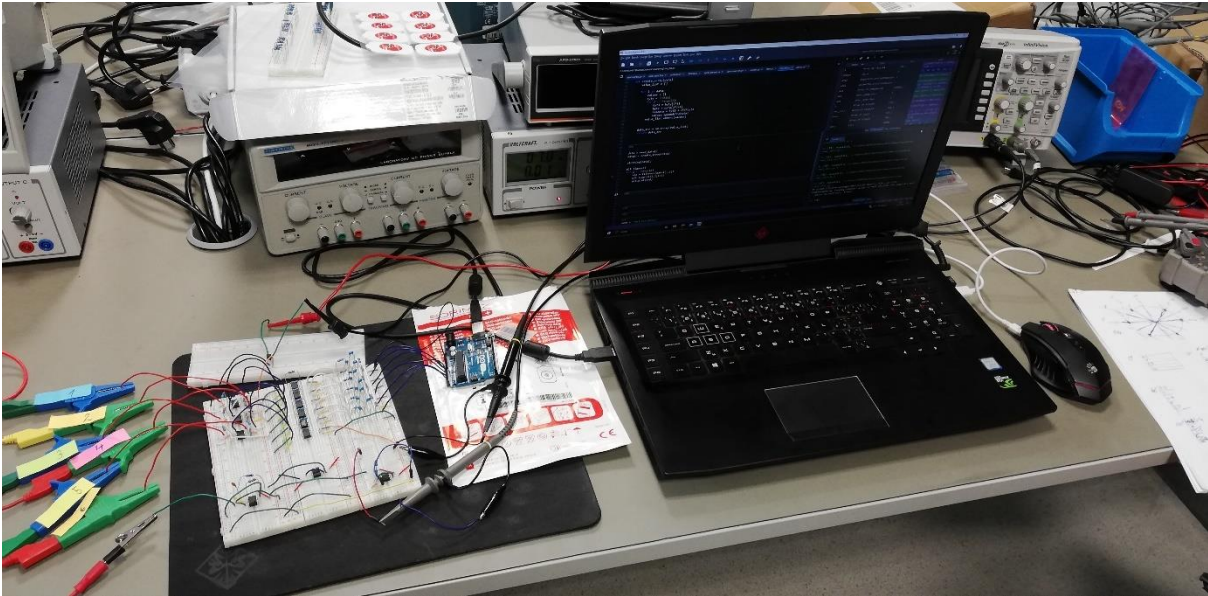
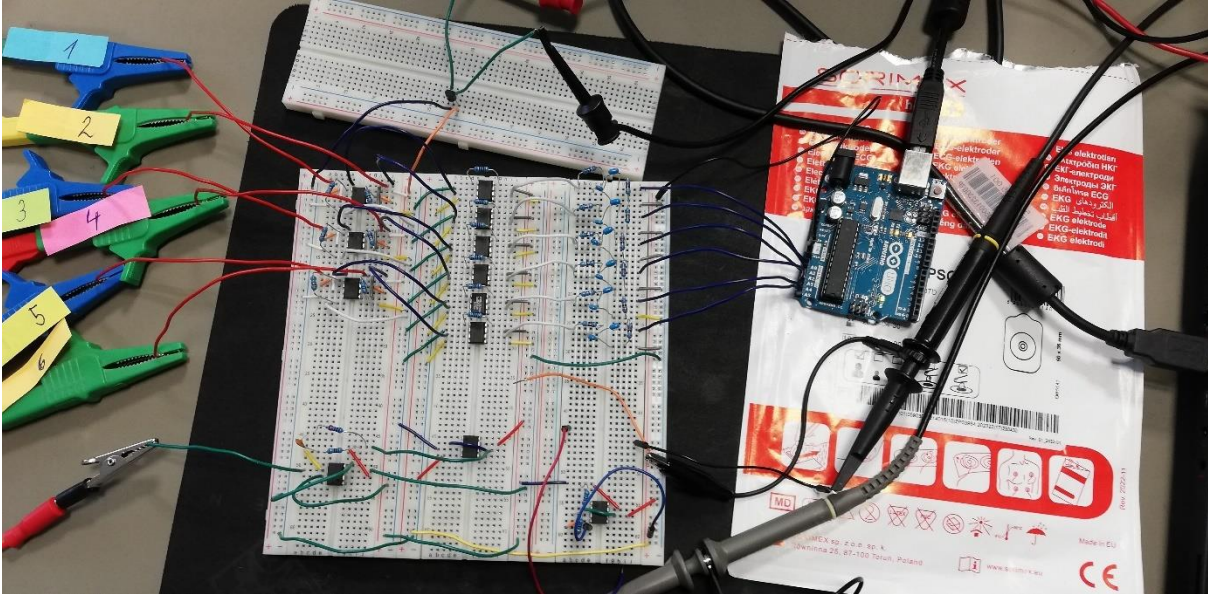
data a=
{analogRead(A0),analogRead(A1),analogRead(A2),analogRead(A3),analogRead(A4),analogRead(A5), count};

Serial.write((byte*)&a, sizeof(a));
```

```
count += 1;
time = micros();

while (time <= (start+492)){
    time = micros();
}
}
```

B: Created measuring circuit



C: Recorded data

Due to extent of the data, only links to digital storage are provided.

Recorded data 1:

<https://drive.google.com/file/d/1FdIKL2jxW0kg2cnTZ1e-x-wPj7PaBEN-/view?usp=sharing>

Recorded data 2:

<https://drive.google.com/file/d/1XhP7b4bR-9-O-yRzdKrzkdFzKA5osVo7/view?usp=sharing>

Reconstruction mask 1 txt:

<https://drive.google.com/file/d/1Hb41alqK-c3r9xWOyKPjMsySYnyuzvZh/view?usp=sharing>

Reconstruction mask 2 txt:

https://drive.google.com/file/d/1h_CaxbiNJTQnX3AJwKfNy5RaERs4z9xq/view?usp=sharing

Reconstruction mask 1 npy:

https://drive.google.com/file/d/1Of5eYVTCbE_MlqHE0poaxWrihyjZiMAH/view?usp=sharing

Reconstruction mask 2 npy:

https://drive.google.com/file/d/1ERqyqJo8ju_5Jf9pgyVfnV2s2oF8EAWA/view?usp=sharing

D: Python code

Due to extent of the code, only links for digital storage are provided.

Main script for program run:

<https://drive.google.com/file/d/1Ydqqpayltvrip9BevKoNrytBV22s9kLw/view?usp=sharing>

Script with all used functions:

https://drive.google.com/file/d/1nM0tYgGBbUqVigzBFhLhIQMBU_99Z20v/view?usp=sharing