Bachelor Graduation Project
Tracheo-esophageal speech enhancement

Input and filter

June 19, 2020

TU Delft

Johan Meyer, 4587103
Folkert de Ronde, 4591852

Abstract

Every year an unfortunate part of the population lose their larynx, often due to the consequences of cancer. The laryngectomy removes the possibility to speak. Fortunately several speech reinstatement techniques have been found. However, all current speech reinstatement techniques do not produce a lot of volume. As a result the laryngectomised person will have difficulties speaking in busy environments, up to the point were the condition can lead to isolation because of the inability to communicate. Moreover with the current methods it costs a lot of energy to speak, the produced speech is less stable and the fundamental frequency has become lower. In this thesis a device is designed to improve the speech produced by a laryngectomised person realtime. The speech is recorded with an electret microphone. The resulting signal is prepared to be quantified to then be processed by a digital signal processing unit. Then the signal is adaptively filtered using the dynamically determined fundamental frequency. By tracking the fundamental frequency some instability from the voice is removed. The fundamental frequency will be determined using the cepstrum, a technique separating the fundamental regency of a voice from the high frequency components added by the oral and nasal cavities during speech. In another thesis a system is designed to amplify the signal to make sure that speaking takes less energy [11]. Furthermore a suitable power source is determined for the system. As the system is designed for Tracheo-Esophageal Speech Signal Amplification, the system will be referred to as TESSA. It is shown that the filter algorithm is capable of successfully finding the fundamental frequency in healthy voices and is stable enough to also find the fundamental frequency in a relatively clear laryngectomised persons voice. It is then proven that the filter algorithm is able to improve intelligibility of noisy voice signals. This is done for healthy voices by taking a clean voice signal and addition of white Gaussian noise.

Preface

This thesis was a work of 2 TU Delft, Electrical Engineering students, a 3rd year and a 4th year student. The thesis was written as the final assignment for completion of the bachelor Electrical Engineering. The subjects multidisciplinary nature and the fact that the research will directly contribute to the improvement of the quality of life of laryngectomised people exited the authors. The thesis is written as a preliminary investigation for a product, eventually to be integrated as a module in a larger system, the EXOBREATHER. Therefore the thesis is written for designers to get a first impression of the scope of the possibilities of the system and a first insight in a possible implementation. Although the thesis is a document in its own right, the research conducted to the system TESSA is done by a collaboration of five students. Two theses are written, each thesis describing a different part of the total system design. This will be further enlightened in the introduction.
We started writing the thesis at the end of April 2020, just as the corona virus arrived and spread largely in Europe. Therefore the original goal to deliver a proof of concept in the form of a physical demonstrator had become impossible, even prohibited. The restructuring of the assignment and the determination of new deliverables went smooth, but still took considerable time. As little experience with the digital nature of the deliverables is between the authors, the determination of them was a challenge.
We would first like to thank the authors of the sister thesis, Bosse, Joris and Tim for their constant collaboration and will to overthink challenges with us. With our close and daily communication this thesis has reached a higher level. We would also like to thank our supervisors ing. R.M.A. van Puffelen and ing. J. Bastemeijer, for our weekly meetings and the guidance that followed from these meetings. Also our client Guus and his friend Peter are to be thanked, because they invested in the project with weekly meetings. These were very useful in guiding the project to useful research for later development. The personal experiences of Guus really motivated us to keep high performance and relate the conducted research to the final application. Then Klaske and her colleague, Ron helped out in starting off in an unknown subject for us, the human voice. With their guidance we quickly acquainted our selves with the basics. Several TU Delft professors helped out by sharing their expertise, therefore we would like to thank dr.ir. G.J.M. Janssen for his help on window techniques and Fourier transform properties and dr. ir. R.C. Hendriks for his experience in speech processing and speech algorithm testing using SIIB.

Contents

1 | Introduction

It is normal for healthy people to be able to make their voice heard during the day. It does not quickly come to mind that there are people who cannot speak as easily as most. This is however the case. Some people are laryngectomised, which means that their larynx and voicebox are removed (a figure of what this looks like is shown in Figure 1.1).These laryngectomised people (LP) can only speak by use of a device.The goal of this thesis is to design a device which makes the LP sound more natural and more intelligible. Note that this thesis is written during the corona outbreak which started in March 2020 in the Netherlands, meaning that the system will not be physically made but the whole thesis will be theoretical, with simulations to make it as realistic as possible.

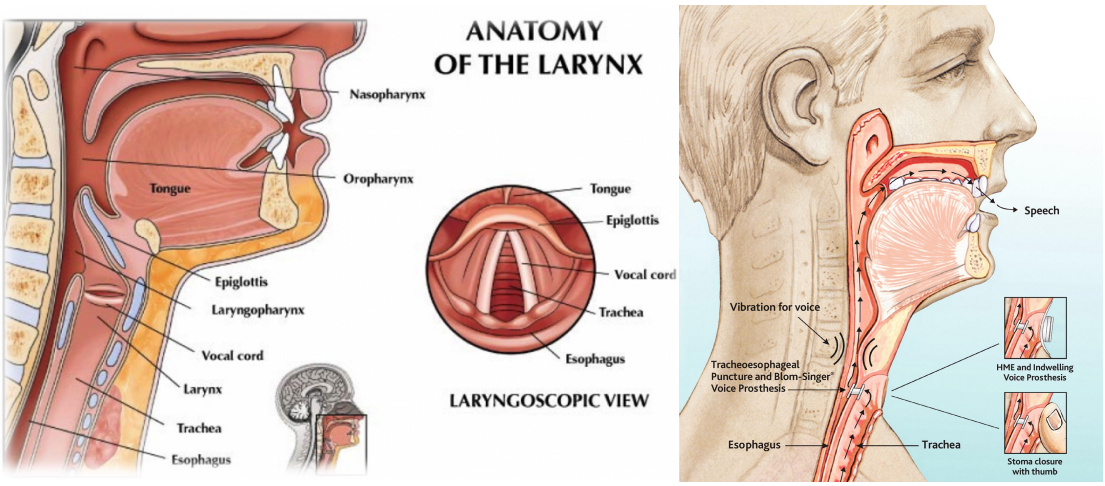


Figure 1.1: Right, the schematic of the anatomy of the Larynx in a healthy person [16] and left, the anatomy of the LP [5]

1.1. Normal speech

This thesis is going to be about speech and thus it is important to first consider how a person who is not laryngectomised speaks. In the larynx the vocal cords are brought into vibration by forcing air from the lungs, through the trachea, passing the vocal cords. The vibrations of the vocal cords give rise to the fundamental frequency (F0). This F0 is then filtered by the oral and nasal cavities. These modifications make it so that the F0 in combination with the movements of the throat result in an audible sound. Sounds produced this way are called voiced sounds. Examples are the 'aaa', 'ooo', 'UUUU'. Another method of producing sound is to rapidly open or close nasal or oral cavities. This way sounds are produced without use of the vocal cords. These are called the unvoiced sounds of which 'p', 't', 'k', are examples [25].
The F0 required to produce voiced sounds has a range differing for men and women as the frequency range is determined by amongst others the length and weight of the vocal cords [25]. A man and a woman typically speak with an F0 in a frequency range from $85-180$ Hz and $165-255$ Hz respectively [3]. The frequency range of the voice is 85 Hz to 4 kHz, although the intelligibility is mostly reliant on the 2 kHz to 4 kHz range [21].
It is also worth considering the time spent speaking on a typical day. A United States teacher speaks 2 hours a day effectively [23]. This is the worst case scenario to be used in this thesis. Furthermore the average person does not speak on maximum volume during the day. For Danish teachers a typical volume profile is given in Figure 1.2. From this figure the average volume level is determined to be at 19.28% of the maximum volume of the speaking person. As the same study shows the teachers speak in a raised voice 61% of the time [15], this estimate should be a sufficient worst case scenario for the calculations in this thesis.

Speech characteristics

Several useful characteristics of human speech that have been defined in the past can be used to evaluate speech. Commonly used are F0, jitter and shimmer. F0 is the fundamental frequency of the voice. The jitter

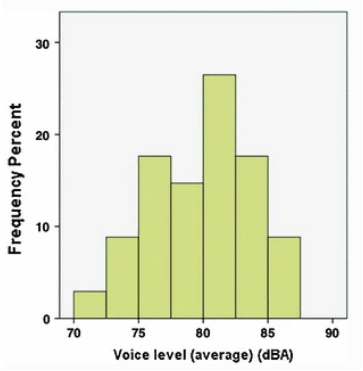


Figure 1.2: Distribution of a teacher daily vocal load during teaching. Measured at shoulder level[15].

is the percentage difference in period between one wave and the previous. The shimmer is the difference in amplitude between successive oscillations [3]. F0 can be determined and shifted, which shifts the harmonics as well, changing the pitch of the voice. The jitter and the shimmer are for now only used diagnostically and modifying them is not in the scope of this thesis.

Model of the human voice

As mentioned before, human speech originates from the vibrations of the vocal cords. A popular model of speech is to take F0 as the input of a filter. The nasal and oral cavities are then approximated by that linear filter. This means that in speech only F0 and its harmonics are present. So speech can be reduced to discrete frequency bands.

1.2. Speech reinstatement methods

First it is important to understand that in a laryngectomy the epiglottis is also removed. The epiglottis closes the trachea when consuming food. As this role can not be replaced at the current state of medical research the larynx and esophagus are permanently separated. Breathing now happens using a stoma in the throat as depicted in Figure 1.1. There are a few ways in which it is possible for LP to speak without a larynx and thus without vocal cords. They could use an electrical device to speak called the electrolarynx. The electrolarynx vibrates the throat tissue by pressing the device to the throat. The F0 is then created by the electrolarynx so the LP can produce speech. However the sound is monotone and is often experienced as robotic. Another option is to train the throat to vibrate the esophagus. For this purpose, part of the esophagus is paralysed. This vibrating replaces the vocal cords and esophageal speech can be produced. The produced voice is however less stable and less loud. The vibrating also takes more effort than the vocal cords would have taken. The third option is to undergo surgery so the trachea is connected to the esophagus again, the tracheo-esophageal puncture. The newly made connection between the trachea and the esophagus needs to be closed with a one way valve. When closing the stoma pressure can be built in the trachea pushing air through the tracheo-esophageal valve causing vibration making it possible to produce tracheo-esophageal speech. [6] As the client uses tracheo-esophageal speech, the rest of the Thesis will be focused on this method pf speech production.

1.3. Laryngectomised speech

Before beginning with the project it is important to know what the difference is in a natural voice and in the voice of an LP. First an LP speaks with an F0 in the range of 50 – 110 Hz, irrespective if the person is a man or woman. Furthermore LP have a jitter (percentage difference in period between one wave and the previous) between 0.61% and 2.6% whereas normal jitter lies below 0.8%. At last LP have a shimmer (difference in amplitude between successive oscillations) typically below 0.5 dB, yet it can be 1.6 dB [3]. Next to the filtering and shifting LP have another problem. They do not speak at the same volume at which a natural voice would be produced. One study showed the maximum volume at which LP can speak is 90.42 dB [20], unfortunately, at what distance was not mentioned. Another study shows that a natural voice can reach about 100 dB [8]. The typical LP however does need to put a lot of effort in speaking loudly making it impossible to speak at a higher volume for prolonged time as this is exhausting.

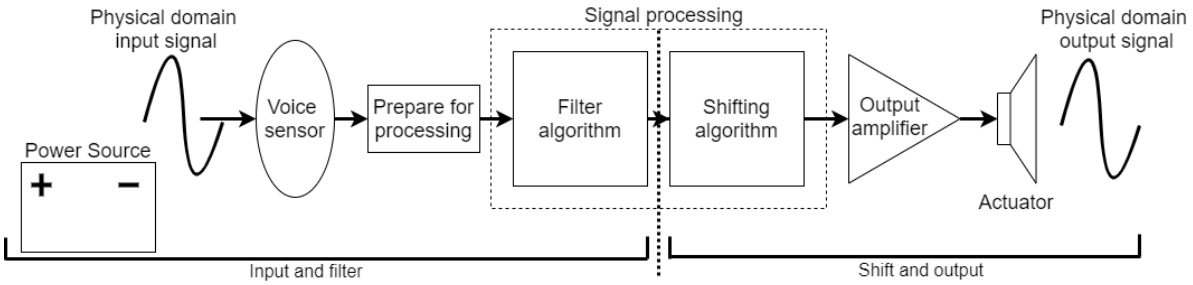


Figure 1.3: The essential system components of TESSA and the division in subsystems discussed in two separate theses. In this thesis the power source, voice sensor and filter algorithm are addressed, in [11] the shifting algorithm, output amplification and transducer or actuator are discussed.

1.4. The system, TESSA

With the information given in section 1.3 it can be concluded that a filter and a shifting device is needed. This is to compensate for the low frequency and the shimmer and jitter. Furthermore it can be concluded that an amplifier is needed to make sure less effort is needed by the user to speak.
The system first needs a sensor to translate the voice data from the physical to the electrical domain. The data is then to be processed to improve on intelligibility as the intelligibility of tracheo-esophageal speech is generally lower than normal speech. This part is thus done by the filter and the shifter. After the signal has been filtered, it needs to be amplified before it is transduced back to the physical domain. Finally the system will need a power source. As the system is designed for use in tracheo-esophageal speech and the main functionality is to amplify the speech signal the system will be named, Tracheo-Espohageal Speech Signal Amplifier, or TESSA. TESSA comprises of a sensor, filter, shifter, amplifier and a transducer. As TESSA is designed by two teams the system is divided over two theses. In this thesis the input and filter mechanisms are designed, tested and discussed as is the power source. The total system and system design division is shown in Figure 1.3.

1.5. Structure of the thesis

First an overview of the required knowledge on the voice and tracheo-esophageal speech was given. In the second chapter the design overview is going to be discussed. In the third chapter the hardware options are discussed. In chapter four a selection is made for every piece of hardware. In chapter five the different parts about the filter design are explained. In chapter six, some limited hardware is designed. Then in the discussion the previous findings is reflected upon and future work is recommended. And finally in the conclusion the reached result is stated.

2 | Design considerations

To align the expectations of the designers and the client a Statement of Requirements (SOR) was agreed upon and can be found in the appendix D. As not all agreements in the SOR are of influence on this thesis, only the relevant agreements are discussed.

2.1. General requirements

The essential selection of requirements from the SOR is stated below:

- The System comprises:
  - A microphone
  - A signal processing part
  - An energy source
- It is wearable on the body
- The input bandwidth is at least 50 Hz to 4 kHz,
- Its operating temperature stays below 36°C,
- The system should provide 2 hours of speaking time.

The product will be designed to improve intelligibility of LP by increasing volume when needed. But as the product should preferably be very small a compromise was reached for the minimum output loudness and size. To keep the size limited but still enable the LP to talk in noisy surroundings the minimum output loudness is agreed upon at 85 dB at 0.3 m. To keep the amplified voice intelligible the bandwidth containing the frequencies of importance in human speech have to be amplified. This bandwidth is found to be 500 Hz to 4 kHz [21]. With the input and output bandwidth at 50 Hz to 4 kHz the frequency range of importance is certainly preserved and also contains the F0. As the product should be wearable on the body it should not heat up to become uncomfortable in use. Therefore the maximum operating temperature is chosen at 36°C which mostly affects the necessary efficiency of all components to keep heat dissipation limited. The speaking time is chosen at 2 hours as it was found that teachers spend about 2 hours per day speaking [23], which can be considered a worst case scenario for this product.

For the microphone, amplifier, speaker and energy source, off the shelf components will be selected. This is done after an extensive selection procedure. The created system must comply to the above requirements. The volume regulation of the system should be managed hands free.

2.2. Signal processing requirements

Additional requirements are set specifically for the necessary signal processing:

- Background noise is suppressed by 6 dB,
- Acoustic feedback should be avoided,
- The F_0 should be found and used for F_0 dependent filtering.
- The voice should be intelligible.

As the product will try to only amplify the sounds of interest and as little noise as possible, the background noise should be suppressed.

2.3. Further preferences

A set of additional preferences were set up to guide the design process:

- The size will have to be as limited as possible
- The total costs of the parts should be around or less than 100 euros
- It is integratable with the 'Exobreather'
- The latency of sound produced should stay below 15 ms
- The reproduced voice should be as natural as possible

The above preferences are guidelines to keep in mind and are mostly topics of future research. These are for now unessential goals as they lay outside the scope of this thesis. For example the component price should be around or below 100 euros to make the product affordable. And as the product records, processes and amplifies the same voice the produced sound of the product could be perceived as an echo due to latency. To prevent this echo from being audible and forming a real hinder the latency should stay below 15ms. Preferably even below 10ms [13]. Ideally the product can perform all above requirements and preferences without losing the natural touch of a voice. And ideally the volume is regulated using the air pressure generated by the LP as in a healthy voice the volume is also regulated by increasing the airflow passing the vocal chords by increasing the air pressure.

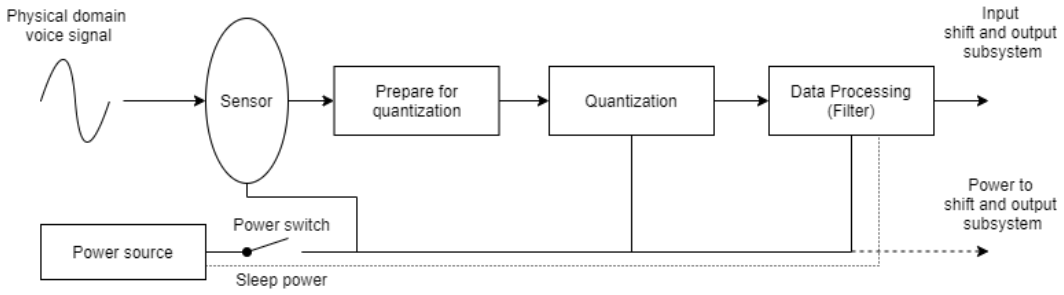


Figure 2.1: A low level block diagram showing the very essence of the project.

With the above requirements and additional preferences a top down, design sequence is started. First the project is brought down to its essence. The result is shown in figure 2.1. This report only focuses on the sensor and the filtering part of the data processing unit.

In the following chapters first the appropriate sensor and actuator are determined to meet the SOR. Then the data processing part is divided in two main functionalities, a filtering and a shifting functionality. Then the desired functionalities of these subsystems and subsequently an appropriate implementation are designed.

3 | Available Technology

To choose the optimal concept for every sub part of TESSA, for every part different concepts are investigated. These concepts are divided in sub solutions and placed in morphological maps. All morphological maps containing the choice process for the optimal concept are shown in Appendix A. For the sensor as an explanation all steps are shown. First all possible concepts are gathered by dividing concepts in subjects. Then for all subjects of the concepts different options are considered. So every morphological map entry is a sub solution of a concept. A concept is then composed by selecting different sub solutions. A typical morphological map is shown for the sensor Figure A.1. To select the optimal sub solutions for the optimal concept, all morphological map entries are rated for relevant categories as shown in Figure A.2. For the switch the weight factors for the categories are calculated by choosing which category is more important then the other to find the weight factors as objectively as possible as shown in Figure A.7. The morphological map entries with the highest score for each subject are then selected. In the figures these sub solutions are outlined in red. From the outlined sub solutions the final concept arises. The result is the optimal concept as depicted in Figure A.3.

3.1. Sensor

3.1.1. Available technology overview

To rate the entries in the morphological map some research is conducted to the relevant technologies. Below different technology types will be discussed.

Dynamic

Dynamic microphones convert sound into an electrical signal by means of electromagnetism. There are 2 types of microphones which are based on this idea. Namely moving coil and Ribbon.

Moving coil

A moving coil microphone consists of a coil which is glued to the rear end of a membrane. There also is a strong magnet surrounding this coil. The membrane moves to the rhythm of the sound waves. In doing so, the coil on the back moves along with it. This moving coil then makes a small signal voltage, because of the relative movement of the coil within its magnetic cap. Now sound has been converted into an electrical signal.

Ribbon

Ribbon work on the same electromagnetic induction principle as moving coil systems do. However this system does not consist of a membrane and a coil. This system consists of a narrow strip of extremely thin aluminium foil. This makes it so that the membrane itself is the electrical conductor. This piece of aluminium ribbon is much lighter than a membrane with a coil of copper wire attached to it. This system is thus more accurate. A downside of this system is that the voltage output is much lower.

Condenser

A condenser microphone consists of a thin membrane also called the diaphragm very close to a metal plate. The diaphragm and metal plate form a capacitor. Due to pressure waves the diaphragm moves and the capacitance changes resulting in a quite high voltage output. Unfortunately the capacitor only stores very little energy so very little current is produced and the signal is therefore quite weak. The electric charge on the capacitor is supplied by an external source.

Electret

The electret microphone is based on a capacitor function like the condenser microphone. However the charge on the capacitor is now statically applied by the 'electret' material. This means the electret microphone does not require an external power source. Also the electret microphone capsules are more mass produced.

Ceramic

The operation of ceramic microphones is based on sound moving a diaphragm. This diaphragm is connected to a piezo-electric material by strut or pin. If the diaphragm moves, the piezo-electric material is deformed by the pin, resulting in producing a varying voltage. Ceramic microphones have internal pre-ampilifiers for boosting the microphone output signal and for producing a low output impedance. These are necessary conditions for their use with transistor amplifiers.

The piezo-electric effect is caused by the molecules of a crystal being randomly distributed. The molecules have positive and negative regions normally resulting in zero polarity. But the charge regions align under stress proportional to the applied pressure. This causes a net nonzero polarity of the material which is measurable.

PZM/boundary

A boundary mic is a type of a condenser microphone.The boundary microphone uses the reflected signal of the surface it is mounted on. Meaning that the diaphragm of the microphone is placed in parallel of the surface it is mounted on. The further functionality is the same as a condenser mic. This makes the microphone more accurate then a normal condenser mic.

Carbon

The carbon microphone is based on carbon (or graphite) granules between a conducting diaphragm and a fixed electrode. Due to sound pressure the diaphragm compresses the granules increasing conductivity. The conductivity then varies proportional to the applied pressure and the sound can be retrieved.

MEMS microphone

Digital and analog integrated circuit microphones exist. Both use air pressure fluctuations to move a membrane with respect to a back plate. The changing capacitance between the membrane and back plate is proportional to the air pressure waves. This capacitance fluctuation can be observed by the output voltage fluctuation. A digital MEMS microphone has a pulse density modulation signal as output. This is comparable to an over sampled signal with a single data path.

3.1.2. Sensor concept choice

After investigating all relevant technologies and based on the findings the morphological map entries are rated as shown in Figure A.2. The optimal concept is found as depicted in shown in Figure A.3 en is a microphone placed near the neck or ear using a clip on system using 2 microphones for surround suppression. Both microphones should be omnidirectional to prevent misalignment. The optimal technology is electret and for flexible implementation an analogue output is preferred.

3.2. Power source

There are different types of power sources. To obtain a good overview of the currently available power sources again a morphological map is made. Some of the relevant available technology is researched to enable a good rating process. All of the batteries considered are safe to use on a human body.

3.2.1. Available technology overview

Li-ion

Li-ion batteries are rechargeable batteries which have a high energy density and a long lifetime. Whilst remaining to have a low self discharge rate. The energy density is equal to 160 Wh/kg. The lifetime is expected to be 2/3 years and the self discharge rate is 5-10% a month. Li-ion batteries usually have a voltage supply of 3.6/3.7 V.

Li-Polymer

The lithium polymer battery has a high energy density as well, but it is a bit more interchangeable. It has a about the same life expectancy as the Li-ion battery and is has a self-discharge rate of only 5%. The Energy density is 100-200 Wh/kg.

NiMH
Nickel metal hydride(NiMH) is a rechargeable battery. The energy density of the NiMH can approach that of a lithium-ion battery while its capacity can be two or three times the capacity of an NiCd battery. These batteries have a higher self-discharge rate then the Li-ion and Li-Polymer batteries. The capacity is 140-300 Wh/L and the self-discharge rate is 10-20% a month.

3.2.2. Power source concept choice
All entries in the map are rated for several aspects each with its own weight factor as shown in Figure A.5. The found optimal sub solutions are then selected and are red outlined in the morphological map Figure A.4. The optimal power source concept is determined to be a rechargeable or replaceable and rechargeable power source as it is practical in use. The optimal chemical turns out to be Lithium Ion and the best charge method is an external charger as is it does not require internal charge circuitry and is flexible in use.

3.3. Switch system
As TESSA only has a job to fulfil when the user wants to speak and as the available energy will be limited, it is worthwhile to limit the power use when the user does not actively speak. For this purpose again a morphological map is made, and depicted in Figure A.6, to envision all possible concepts. Then all morphological map entries are rated for their switch speed, the effort needed from the user to switch on. The notability of the switch depending on location and size of the technology. The naturality of the switching action (which coincides largely with the effort parameter). And finally the power needed when the switch is in off state, so if for example constant active sensing is required. By comparing these weight categories the actual weight for each category is determined relative to the other weights. The determination of these weights and the rating of the morphological map entries are shown in Figure A.5.
The best concept is found to be a physical switch controlled by a button or an air pressure guided system. And the switch should be either integrated in the casing of TESSA or be placed near the stoma on the throat.

4 | Selection of hardware

4.1. Microphone

Specification

After completing a morphological map taking into account all design considerations it became apparent a small microphone should be chosen with a high flexibility in placement. This way at a later stage the microphone location can be optimised to increase customer satisfaction. Due to the dependability of the frequency characteristic on the location of microphone placement [21], the microphone needs an as flat as possible frequency response. Many off the shelf microphones have a frequency response tweaked to the intended placement. Also to keep placement flexible the microphone should be omnidirectional. The microphone should be able to record the human voice in its relevant frequency range. And the frequency range containing the important information to ensure intelligibility is 500 Hz to 4 kHz [21]. Finally the sensitivity of the microphone should be as high as possible just as the signal to noise ratio.

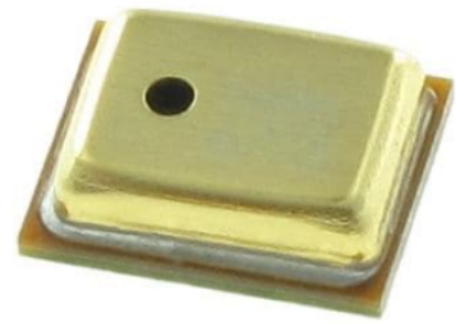
Selection

Multiple microphones were selected according to the specifications. These microphones were rated for their respective qualities on bandwidth, sensitivity, cost, directionality and signal to noise ratio. From the most promising microphones an off the shelf microphone was selected as it is already integrated in a casing and is well suited for flexible placing. Also a MEMS microphone was selected. This microphone will have to be supplied with a readout circuit and in development a casing. The MEMS microphone was selected for its more extensive available product information and its significantly higher sensitivity compared to the other microphones. The microphone and its key performance parameters, found in the data sheet [14], are shown in Figure 4.1.

4.2. Battery

Specification

To make sure the system will work for a whole day, it is important to know the power (draw by the circuit), the amount of time someone is speaking during an average day and the volume on which someone speaks in a day. The power of the circuit is estimated to be relatable to the power required at the output. So at most 1.5W for the speaker as most 1.5W speakers will be able to deliver the required output volume. Then the amplifier will deliver this power but will have a loss estimated at an efficiency factor of 0.8. It is then estimated that the DSP will consume comparable or more likely less power than the amplifier, so an equal power consumption if estimated.
Then the estimated speaker dissipation is 1.5W of electrical energy into sound, the amplifier is then estimated to dissipate $(1.5/0.8) - 1.5 = 0.375$W into heat and the DSP dissipates $(1.5/0.8) - 1.5 = 0.375$W into heat. This sums the power of the system to $1.5+0.375+0.375 = 2.25$W peak power. The DSP power budget was then compared to several actual DSPs and the estimation was deemed realistic. These values are the power budgets for



Microphone Parameters	
Series name:	SPU0414HR5H-SB
Technology:	MEMS Electret
Signal type:	Analogue
Dimensions:	3.76*2.95*1.1 mm
Operational bandwidth:	100Hz -10 kHz
Sensitivity:	-22 dB
SNR:	59 dB
Impedance:	400 ohm
Typical price:	€1,52

Figure 4.1: The chosen microphone with its parameters.

Component	Power budget
Speaker	1.5W
Amplifier	0.375W
DSP	0.375W
Complete system	2.25W

Table 4.1: Power budgets for the electrical components of TESSA using the most power

the system and are again listed in Table 4.1. It is estimated that the microphone and input circuitry (anti aliasing filter and the like) use negligible or no power from the battery. After the maximum consumption of the system is calculated, the required energy to make TESSA last an entire day can be determined. The amount of time someone speaks greatly depends on the lifestyle of a person. Therefore the worst case scenario is taken. Namely the lifestyle of a teacher. For the amount of speaking time, 2 hours is taken as it is shown that teachers speak for 2 hours a day [23] as was mentioned in section 1.1. The last variable to be determined is the volume at which someone speaks which was done in section 1.1 as well, using [15]. The average speaking percentage of the maximum voice loudness is calculated at Vol_{avg} = 19.28% of the maximum volume. All these variables are taken into consideration to calculate the needed capacitance of the battery. Next to this there is also a 20% safety margin taken into account. To make sure that the battery does not deplete before it's promised lifetime. The formula to calculate the battery capacity is given in formula Equation 4.1.

$$C_{Bat} = \frac{(((P_{speaker} + P_{amplifier}) \cdot Vol_{avg}) + P_{DSP}) \cdot t \cdot 1000}{V_{Bat}} \cdot F_{Safety} \tag{4.1}$$

The equation parameters are described below:

- C_{Bat}: Battery capacity in mAh
- $P_{speaker}$: Power of the speaker in Watts (1.5W)
- $P_{amplifier}$: Estimate amplifier dissipated power dependant on output power (0.375W)
- P_{DSP}: Estimate DSP dissipated power(0.468W)
- Vol_{avg}: Average volume on which a person speaks in comparisment with the maximum volume (19.28%)
- t: Amount of time a person speaks a day in hours (2h)
- F_{Safety}: Battery life safety margin (20% = a factor of 1.2)
- V_{Bat}: Battery rated voltage in volts (3.7V)

To illustrate, the formula is filled in: $C_{Bat} = \frac{(((1.5+0.375)\cdot 0.1928)+0.375)\cdot 2\cdot 1000}{3.7} \cdot 1.2 = 477.719$mAh. With this formula it is determined that for a 1.5 Power level for the circuit, an 80% efficiency for the DSP and the amplifier and a 20% safety margin that the capacity needs to be about 500 mAh at 3.7V.

With the found power budget the minimally required discharge current for the battery can be calculated using Equation 4.3.

$$I_{PeakDischarge} = \frac{P_{peak}}{V_{Bat}} \tag{4.2}$$

It is found that the peak discharge current should minimally be 2.25W/3.7V = 0.608 Ampere. And for the typical discharge current the average power use is calculated using Equation 4.3.

$$I_{TypicalDischarge} = \frac{((P_{speaker} + P_{amplifier}) * Vol_{avg}) + P_{DSP}}{V_{Bat}} \tag{4.3}$$

The typical discharge current should be around $(((1.5 + 0.375) * Vol_{avg}) + 0.375)$W/3.7V = 0.199 Ampere. The battery chemical is preferred to be Li-ion because this battery technology has a high energy density and a long lifetime.

Battery Parameters	
Series name:	DRJ3555
Chemical type:	Li-ion
Rated voltage:	3,7 V
Typical discharge current:	98 mA
Maximum discharge current:	980 mA
Charging current:	245 mA
Internal resistance:	200 mΩ
Capacity:	500 mAh
Energy:	1850 mWh
Dimensions:	35,2*5,7 mm
Volume:	5547 mm^3
Energy per volume:	0,334 mWh/mm^3
Price:	€28,00

Figure 4.2: The chosen battery with its parameters [18].

Selection

Several batteries are gathered, all selected according to the concept selected using the morphological map as described in section 3.2. The selected batteries are then rated for their capacity, stored energy, volume, energy per volume and cost. The capability to store energy is rated double by both rating capacity and energy. Also energy and volume are rated double by rating energy per volume again. This is justified as these measures are very important in the selection. The price of the battery turned out to be of no influence in the rating system which is justified as all prices of the selected batteries are affordable. But in more production based design this might change. The rating system filled in and with results is given in Figure B.1. The selected battery is the one with the highest score which is the RDJ3555 rechargeable Button-cell Li-ion battery. The battery and its main parameters, gathered from the data sheet [18], are depicted in Figure 4.2.
With an rated voltage of 3.7V it is easily integrated in the system. The typical discharge current is rather low but the maximum discharge current is easily sufficient. The capacity of 500mAh is sufficient and the dimensions are still small so the battery is easily integrated in TESSA. With a charging current of 245mA the battery can be charged in 500mAh/245mA ≈ 2 hours.
This is not a standardised lithium ion button-cell, but the operating voltage and battery technology is standard. Therefore an off the shelf charger can be chosen, this falls out of the scope of this thesis. Finally the price is rather high but the small size and high capacity easily outweigh this factor.

4.3. Digital signal processing unit

The signal processing unit is chosen to be an Digital signal processor (DSP). This is because the system requires calculations to be done in the frequency domain, resulting in the need for an FFT. An FFT can be done in hardware, using an FPGA, but it can also be done by a DSP. For development flexibility a DSP is chosen to later implement software implementations on. More on the choice of DSP model, its operational parameters and cirquit implementation can be found in [11].
This design requires a DSP that is small and power efficient. It should also be suitable for audio applications and be able to perform operations on audio signals in real-time. In a later stage of design it is desirable to have a convenient testing environment. A software suite and demonstrator board to accommodate this need are nice-to-haves. A cheap and readily available model is chosen in Texas Instruments' C5000 family of chips. Texas Instruments works in close cooperation with the TU Delft which provides ease of access to samples and support. In addition, the Code Composer Studio (CCS) is a free software suite available on the Texas Instruments website and will be suitable for testing the design later on. This is all more elaborately discussed in [11].

5 | Filtering

At the heart of TESSA is the filter algorithm described in this chapter. To improve the intelligibility of the recorded speech signal before amplification at the output of TESSA, the undesired data should be eliminated. This is all done in the upcoming filter algorithm. First the desired filter functionalities are summed up and an appropriate filter method is chosen for all desired filter functionalities. The technical theory is discussed in this chapter as well. Then the implementation of all functionalities are discussed. Then the functionalities are tested and the results are presented.

5.1. Functionalities and method

To design an algorithm which will function as desired, first different filtering methods were considered. Then different approaches to data segmenting through windowing were investigated. The filter functionalities were chosen for their ability to improve speech quality.

5.1.1. Low power filter

The incoming audio data can be reduced by an amplitude filter. This increases the data quality because this makes sure that background noise is reduced even before going to the frequency spectrum. The amplitude filter checks if there is significant signal power in a set of samples. If this is not the case, these samples do not contain information and must be noise. Therefore the amplitude of those samples is set to 0 and the samples are marked as noise. Which means that the device does not start playing if the amplitude filter recognises something as noise.

5.1.2. Fast Fourier Transform

To process the audio from the microphone in the frequency domain the discrete time signal is converted to a discrete frequency signal using the DFT. The Fast Fourier Transform (FFT) is a very fast algorithm to implement the DFT and was described by Cooley and Tukey in their paper [12]. Using the FFT algorithm, the DFT can be preformed faster than originally was the case. They brought down the computation time from N2 to Nlog(N) [1]. This paper, by Cooley and Tukey, shows how to implement the radix-2 FFT. For the radix-2 FFT the vector length N should be a power of 2. This is because the FFT only takes storage locations which are multiples of 2 [12]. The FFT algorithm of MATLAB therefore operates much faster if the FFT is of a size which is a multiple of 2.

5.1.3. Windowing

In order to process data digitally the data is segmented. To reduce the negative effects of segmenting a windowing technique is used. A window technique takes a signal and breaks it down into sine waves of different amplitudes and frequencies [10]. This is important because an FFT is used to transform our time signal into a frequency spectrum. The FFT transform assumes that the data set is a finite data set which has a continuous spectrum that is one period of a periodic signal. This is however not the case. This results in a truncated waveform with different characteristics from the original continuous-time signal. The spectrum will show an enormous peak in the frequency domain, which are not present in the original signal. This phenomenon is called spectral leakage [10]. To make sure that these frequency components will not be in the audio sample, a windowing technique is used. The required window will need to overlap several data points with two frames making it so that all data points are considered and correlated with the adjacent data points. In speech analyses it is needed for every data point to be considered in correlation to the adjacent datapoints because the amplitude difference gives information about frequency. A window is a multiplication factor, which has the property of having zero output (multiplication) outside of the window and thus only having a value inside the window (with a maximum in the middle). A windowing technique reduces the amplitude of the discontinuities at the boundaries of each finite sequence. To make sure every data point is taken into consideration and is taken in correlation with its adjacent data points, these maxima must have equal distance to each other.

5.1.4. Noise reduction filter

The system should be able to reduce the noise as well. To do this, different types of filters are considered, namely the wiener filter and the F0 dependent filter.

Wiener filter

First the Wiener filter was considered, which takes the distorted signal and estimates the magnitude spectrum of the undistorted, noiseless signal. It does so by applying an adaptive filter on the noisy signal. The filter is updated depending on the constantly updated estimates of the noise and signal spectra [2]. This filter thus needs a microphone which only catches the noise in order to substract the noise from the original signal. A few problems with this filter are the fact that this noise is never exactly the same for both microphones and the fact that the 'noise microphone' also catches the speakers voice, resulting in the loss of information.

F0 dependent filter

F0 filtering is based on the principle of how human speech works. Therefore first, a brief recollection of section 1.1. Our speech is formed by a fundamental frequency (F0) initiated by the vocal cords. This fundamental frequency is modulated mainly inside our oral cavity and nasal cavity, the vocal tract. Meaning that the vocal tract approximately functions as a linear filter on F0. The F0 implementation is a filter in which the fundamental frequency is adaptively determined as it varies continuously during natural speech. Then F0 is used to find all harmonics in the speech signal which are integer multiples of F0. Then all frequency components from the signal spectrum are removed that are not harmonics. To filter using F0, first F0 needs to be determined. Several techniques are compared.
Another thing to take into mind when developing a filter which is F0 dependent is that the determined F0 can also be used to reconstruct speech [24].

Cepstral analysis

The first way to find F0 is cepstral analysis. The cepstrum is the inverse Fourier transform of the natural logarithm of the Fourier transform of the time domain signal. It is thus an inverse Fourier transform of the logarithm of the frequency spectrum. The equation can be seen in Equation 5.1.

$$c = IDFT(log(|(E(\omega)|) + log(|(H(\omega)|)) \tag{5.1}$$

In this equation c is the cepstral spectrum called the quefruency [17], E is the fundamental frequency and H the changes made to the signal by the mouth represented by a frequency filter [17]. This equation origins from the fact that the audio output of our mouth is made by use of transforming the fundamental frequency. This means that the audio output can be represented by the fundamental frequency multiplied by a transfer function (filter). If this equation is put in logarithmic values, the fundamental frequency and the filter will add up instead of multiply, resulting in the separation of the both. The fundamental frequency will only have influence on the low frequencies while the filter will only have influence on the high values of the frequency. The datapoints are given in ms and the cepstrum returns peaks on which the fundamental frequency is found.

Normalized autocorrelation function

This function determines the Fundamental frequency by use of a windowless normalized autocorrelation function (ACF). A benefit for this kind of function is the fact that this function is immune for noise [7]. This function however has difficulty determining the fundamental frequency for higher pitch signals [7]. A second problem with this function is the fact that a clear peak can be missed in a formant structure [7].

F0 tracking

As F0 in a tracheo-esophageal voice is less stable some algorithm will need to track F0 so when an unlikely value is found the filter bases its parameters on a previously found F0. The F0 tracker thus makes sure that the stability of a clean voice is realised by use of previous F0 values.

Zero padding

In order to make sure the FFT is done as fast as possible, the samplesize should be a power of 2. This has already been mentioned in subsection 5.1.2. To make sure every data set has a samplesize which is equal to a power of 2, zero padding has been used. The data set gets zeropadded to the nearest next power of 2. In FFT the frequency resolution is $\Delta R = f_s / N_{fft}$ inclusive zero padding [9]. Time domain resolution is $\Delta R = 1/T$ [9].

5.1.5. Overlap and add

As has been mentioned in section subsection 5.1.3, the audio data is cut in halve and halve of the audio data is windowed by an windowing function. The other halve is then windowed by another windowing function. In order to get the original data back, the overlapping data of the windowing functions needs to be added up. Both windowing functions added up with each other will result in a multiplication factor of 1 again. Which means that the original data is again represented. The manner in which this works is depicted in Figure 5.1. As can be seen in this figure, when the window functions are added together, it represents a stable signal.

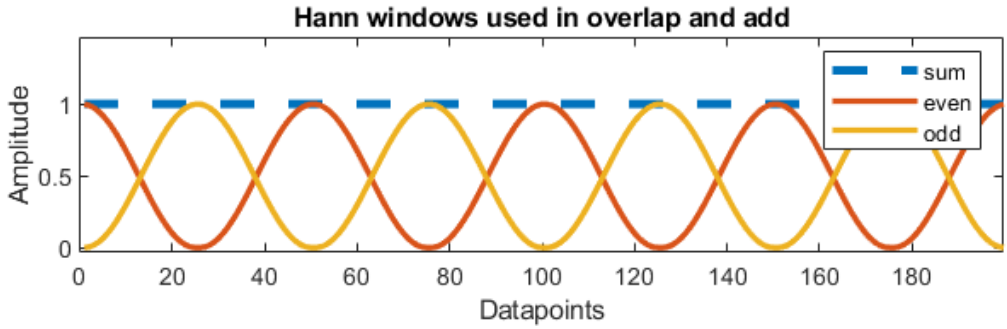


Figure 5.1: An example of what the overlap and add of the windowing function looks like. In case of a Hann window and a 50% overlap, the sum of two consecutive windows (an even and odd index) sum to an amplitude of 1.

5.2. Implementation

The algorithm written to make the filter work actually consists of multiple functions which work together to make a properly working filter. An overview of this system can be seen in figure 5.2. All the different components will be discussed in the following subsections.

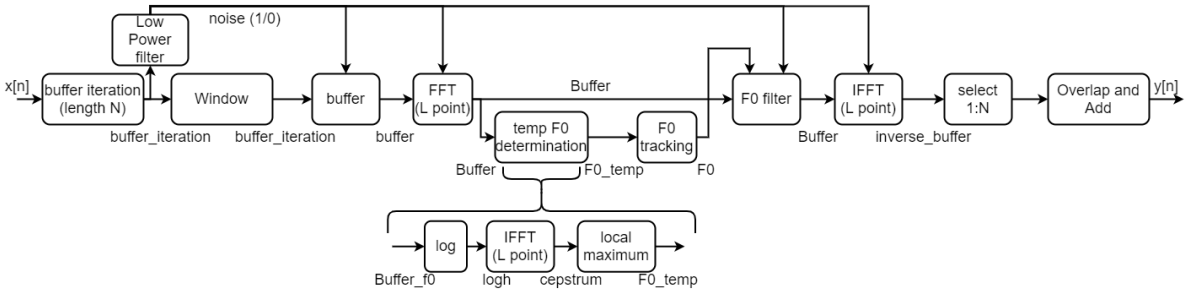


Figure 5.2: Filter system overview showing the data path and all operations performed on the data.

5.2.1. Iteration buffer

As the input data is read in and it is stored in the iteration buffer. This buffer size is limited by the maximum allowable latency. As mentioned before, this is about 10ms. A longer buffer means a higher frequency resolution, but also increases latency.

After the filter functionalities were completed, a test was conducted in which the latency was set to different values. Then the SIIB algorithm was used to envision the performance of the filtering algorithm with different latencies for the iteration buffer. This SIIB algorithm is an algorithm written to find the influence on the intelligibility of an algorithm by comparing a clean signal with the signal altered by the algorithm [19]. SIIB can then be used to compare the performance of different algorithms working on speech signals by comparing their relative influence on the intelligibility found by SIIB.

This test was done for multiple letters. The graph of this test can be seen in Figure 5.3. With this graph it is proven the latency of 10ms is the smallest latency before the filtering algorithm intelligibillity result really deteriorates. A longer latency would be better, but this is still limited by the desired system latency.

5.2.2. Window

The window has been implemented for different types of windows. Namely the hamming, hann and rectangular window have been implemented and tested against each other. The windowing techniques have been

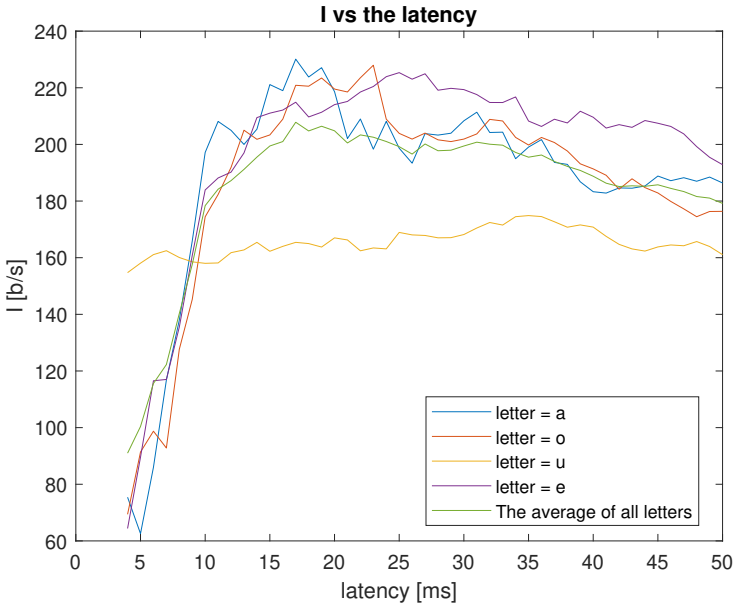


Figure 5.3: I vs the latency.

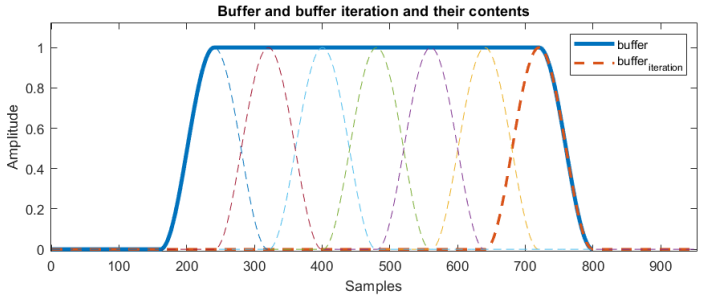


Figure 5.4: Contents of the buffer, the iteration buffer currently being added to the buffer and the overlapping iteration frames currently in the buffer.

implemented as a function in which the type of window can specified in the function call.

5.2.3. Low power filter

The low power filter implementation is an easy algorithm. It checks for every 100 datapoints if the sum of the datapoints is greater then a certain threshold. If this is the case then the datapoints are considered to be noise and therefore their amplitude is set to 0 and the data points are flagged as being noise. When the code sees the fact that data points are flagged, it does not process the data anymore, resulting in no output. This filter thus does nothing when the user is actually speaking.

5.2.4. Buffer

The buffer is needed in order to increase the accuracy of the F0 determination algorithm. The F0 determination algorithm was determined to be accurate if it was given 40 ms of samples. This is too much latency to be able to avoid the creation of an echo since a latency over 15 ms creates an audible echo [13]. To ensure the fact that the F0 determination algorithm works properly, a buffer is made which adds the last recorded iteration buffer to the larger buffer vector and removes the iteration buffer data which was recorded 40 ms ago. As the windowed iteration buffers are overlapped 50% the buffer stores 7 overlapping data sets of 10 ms. This buffer ensures that the system still determines fundamental frequency every 10 ms and it processes this data in the same time-slot, yet using 40 ms to remain accurate. Figure 5.4 shows how the buffer is filled.

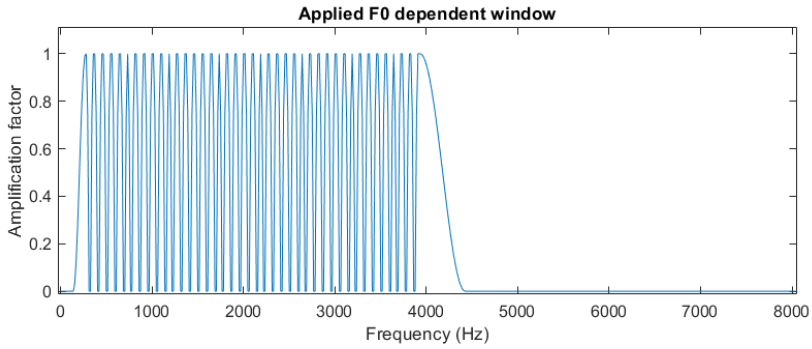


Figure 5.5: Applied window filter, suppressing all frequency components that are not harmonics and suppressing the first 5 harmonics and all frequency components over 4kHz

5.2.5. Temporary F0 determination

The temporary F0 determination function estimates the current F0, this F0 is later checked by the F0 tracking algorithm. The temporary F0 determination function determines the F0 by use of the cepstrum. The cepstrum had been chosen because of the fact that the cepstrum is accurate for 'high' pitches as well. This product must work for woman as well, thus it is important to be able to determine 'high' pitches. The normalized autocorrelation function could not have been as accurate as the cepstrum in this case. The cepstrum takes the inverse Fourier transform of the logarithm of the frequency data and then the code checks where the maximum lays in a specified frequency range. Due to harmonics, this frequency range is different for women and man. For man the frequency range is specified as 50-100 Hz and for woman it is specified as 150-200 Hz. The code finds the peak inside these frequency ranges and then returns F0.

F0 tracking

To make sure that the code will be even more accurate, a piece of tracker code is implemented. This code checks if the fundamental frequency does not change with a higher value then 10 Hz. This range of 10 Hz is chosen because the fundamental frequency changes with about 10 Hz max in speech. At last the code also takes into consideration that one might speak with an intonation. This means that the code will check if someone has changed the frequency of their voice, resulting in a correct change in the fundamental frequency. In Figure 5.13 an audio sample is used in which an a is spoken out and it is tried to hold that a at the same frequency. In this figure it can be seen that there is an outlier, which has been notified by the tracking algorithm and has been ignored. In Figure 5.14 an audio sample is used in which a lot of different frequencies are used. It can be seen that the frequencies do change a lot and it can as well be seen that if there is an outlier determined for more then 3 times that the code jumps to the new F0.

5.2.6. F0 dependent filter

The F0 dependent filter is actually a bandpass filter which only passes a certain frequency range around a given harmonic of the fundamental frequency. The F0 dependent filter thus uses the determined F0 for the sequence, it then uses this F0 to determine it's harmonics. Around these harmonics, the bandpass filter is called. The first 5 harmonics are filtered away because of their non-importance for the intelligibility. This means that the first peak which is not filtered is around the 500 Hz. It has as well been afformentioned that a maximum of 4 kHz is needed for intelligibility, thus after 4 kHz every part of the spectrum is filtered. The actual applied filter is then shown in Figure 5.5.

Bandpass filter

The bandpass filter is actually a bandstop filter which is called by the F0 dependent filter function in a smart way. The bandstop filter is called to filter every value around the harmonics. After the fundamental frequency is determined, the harmonics up to 8 kHz are determined and stored in an array. In section 1.1 a maximum of 4 KHz was determined. Using this array an adaptive bandpass filter is called to apply a window in the frequency domain. Around every harmonic several bins are perserved. On the other bins the window is applied. This window is dynamically determined for each attenuated band. The SIIB algorithm [19] is used to determine the optimal window length and perserved frequency bins around the harmonics to maximise intelligibility.

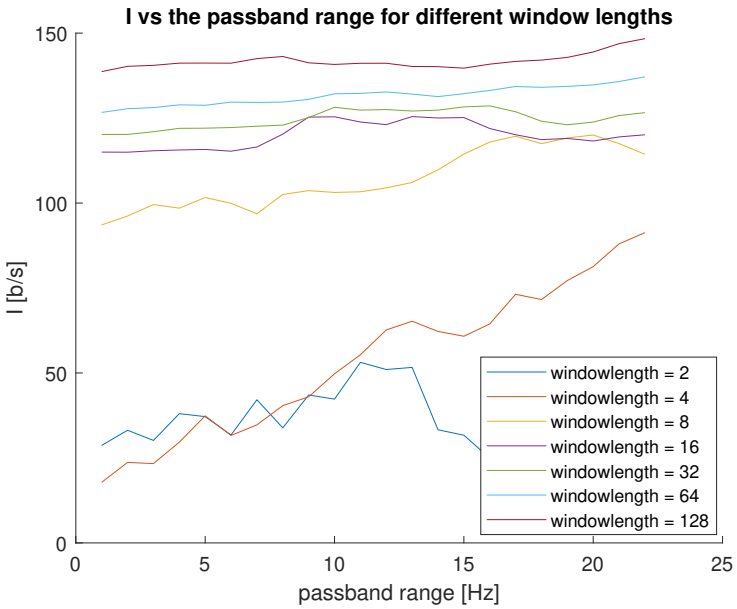


Figure 5.6: I vs the filter range for different window lengths

5.2.7. Overlap and add

The overlap and add functionality is performed by simply selecting the data of one iteration from the buffer and return that data in the datastream. Then in the next iteration the buffer is shifted half a framesize and is updated with a new windowed frame. At the output a framesize is again selected and added to the datastream which has also moved on half a framesize. The applied Hann window then causes the datastream signal to add up to a constant signal.

5.3. Testing

To ensure the functionality of the filter, the filter needs to be tested. This is done for every function and for the system as a whole.

5.3.1. Window and buffer

To ensure the correct functioning of the iteration buffer, the window and the buffer a 200Hz sine is examined. The samples of the sine are first loaded in the iteration buffer, then multiplied by the time window. The resulting sine is plotted. Then the windowed samples from the sine are placed in the buffer using overlap and add. The resulting filled buffer is plotted as well. The use of a very simple signal quickly visualises any errors in the implementation.
After the signal is processed at the end of the filtering algorithm, after ifft, the buffer is read out and the data stream outside the algorithm is reconstructed using overlap and add. To test if this functionality is functioning up to expectations the time domain signal of the put back samples of a 200Hz sine are plotted. The data is undergoing an fft and ifft but no filters are applied.

5.3.2. Low power filter testing

The low power filter is tested in an easy way. The original signal and the signal after the amplitude filter are played after each other. The difference in the audio signals was easy to establish, meaning that in the amplitude filter there was no background noise when nobody spoke. There has been a graph made as well in which the amplitude of the power can be seen, with respect to the old power output at that point in time.

5.3.3. F0 determination testing

The F0 algorithm is tested by use of a recording of our own voice. Meaning that we already know the fundamental frequency of our voice by determining it by hand. The fundamental frequency is then determined by use of the algorithm and compared to the algorithm determined by hand. These values were 83 and 81 Hz. Since the chances are quite high that our determination by hand was slightly off. this algorithm is taken to be working correctly.

5.3.4. Filter testing

At first the filter was tested being an bandstop filter. The only thing the filter did was setting certain frequencies equal to zero. Then the implementation which was dependent of F0 was tested. To test this, the frequency data of a healthy voice was taken. To this data noise was added and then the data was filtered.

5.3.5. Whole system testing

The whole system needs to be tested as well. This is tested for a full audio document, but this is not the same as how it will be in real life, since in real life the sample size will be much smaller due to latency issues. Thus the whole system is tested in two ways. First the whole filter is tested with a full audio fragment and then the system is tested with the audio fragment cut into small fragments and put together. The system is cut into a samples of a samplesize equal to the latency multiplied with the fundamental frequency. The actual testing is done by use of the SIIB algorithm [19]. The SIIB algorithm returns the amount of bits per second (I) which correspond to the clean version of speech. The SIIB algorithm thus needs a sample of clean speech, therefore our algorithm is tested by use of an audio fragment which has been made by one of the authors, since the author's voice is clean. This audio fragment is then filled with noise. After that the audio fragment is filtered and the resulting I is determined. In this way the functionality of the algorithm can be tested. The algorithm makes use of multiple variables which can be set. To determine the best choice for the variables, a testing algorithm is written. This algorithm includes 2 for loops. 1 in which the filter range is set from 1 to 22 Hz and 1 in which the window length is set from 2 to 128 with only taking powers of 2 into account. For all these variable settings, an audio file is filtered and the I is determined. This I is then set into a graph and in this way the best values for the variables can be determined.

5.4. Results

After a testing plan was described for each filter functionality, the results are presented next.

5.4.1. Low power filter

As can be seen in Figure 5.7, the power of the parts which were considered to be noise have a value set to 0. The low power filter thus does what it needs to do in this aspect. The second aspect which was going to be tested is the fact that all these values are flagged to be noise. This is tested by use of the fundamental frequency testing. In Figure 5.8 it can be seen that the fundamental frequency up until the speaking started is equal to 0, with 1 outlier. In both the diagrams it can be seen that in this outlier there actually was a signal which was captured. The system thus works exactly as it is intented to do.

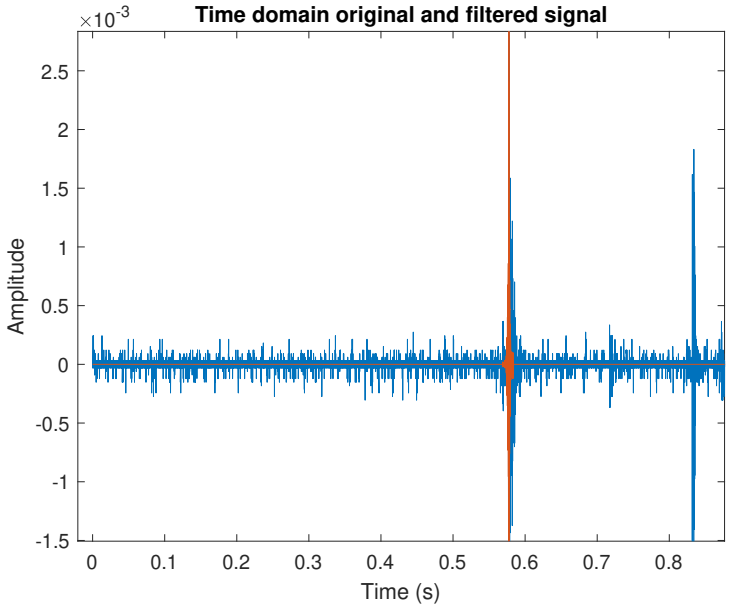


Figure 5.7: The low power filter of an noise signal with a short sound.

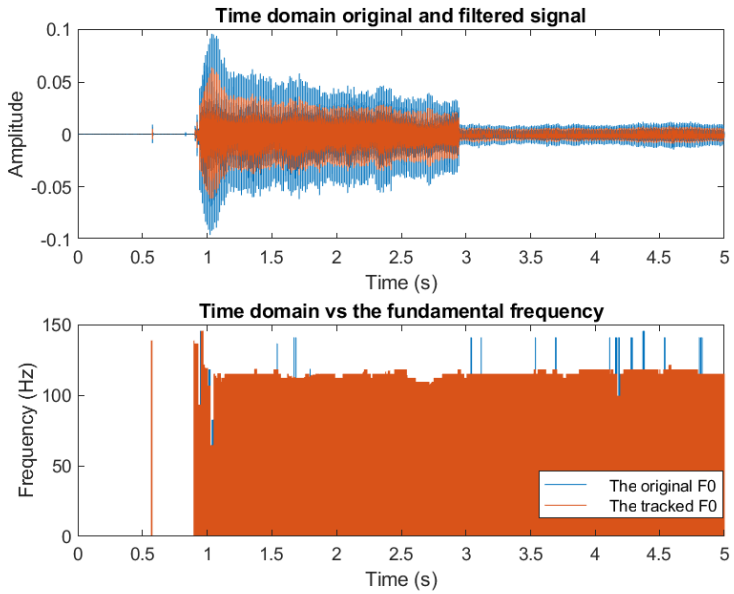


Figure 5.8: F0 filter, with noise detection system, measured with a sample rate of 44.1 kHz while speaking the letter 'a'.

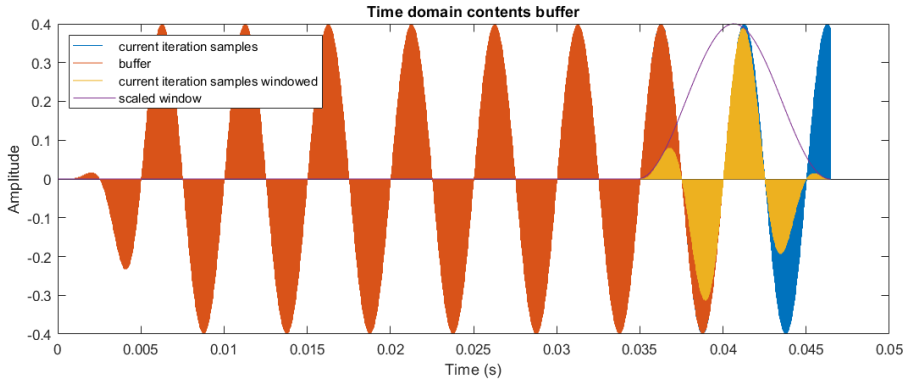


Figure 5.9: Contents of the buffer, the samples currently being added to the buffer, the samples currently added to the buffer windowed and the Hann window itself. The window is scaled to fit in the figure, its maximum is at 1. Using overlap and add the buffer is filled. Here the buffer is completely filled using a 200 Hz signal for envisioning the functionality.

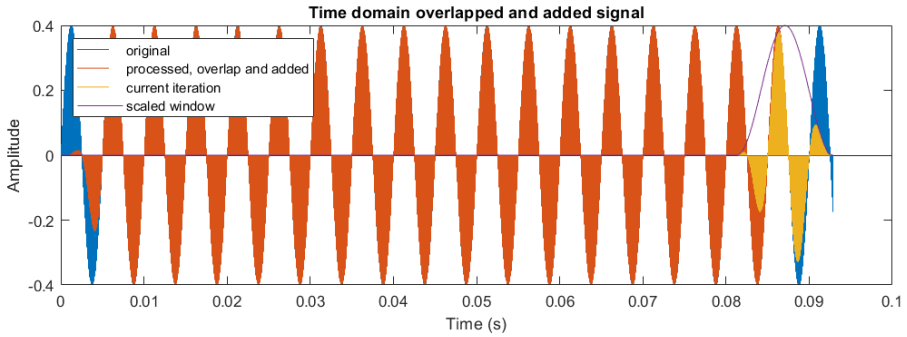


Figure 5.10: The data after fft and ifft put back in the continuous data stream using overlap and add in steps of half window lengths. Blue is the original data, orange the data put back and yellow is the currently being put back data. The applied Hann window is scaled to fit in view, its actual maximum is at 1. Only the first samples are not restored completely using this window overlap and add technique.

5.4.2. Window and buffer

The simple 200Hz signal test results in Figure 5.9, depicting the filled iteration buffer with the applied window. The window is applied well as the iteration buffer contents nicely follows the window shape. The filled buffer is shown and proven to function properly as the sine wave in the buffer corresponds exactly to the original sine. The reconstructed data of a 200Hz sine after fft and ifft, but no filters applied, is visible in Figure 5.10. The orange data is the data returned in the data stream. As the orange data overlaps the original data depicted in blue completely the data is not deformed by the buffer so the buffer and the overlap and add function properly.

5.4.3. F0 determination

The F0 determination function is first tested by making a graph of the cepstrum and determining by hand where the maximum should be. Also it is checked if the maximum is indeed clearly indistinguishable. The graph can be seen in Figure 5.12. In this figure it can be seen that there is a clear peak and it can be seen at what time this peak originated. With this time, the frequency can be determined. This particular simulation resulted in a frequency of 83 Hz.

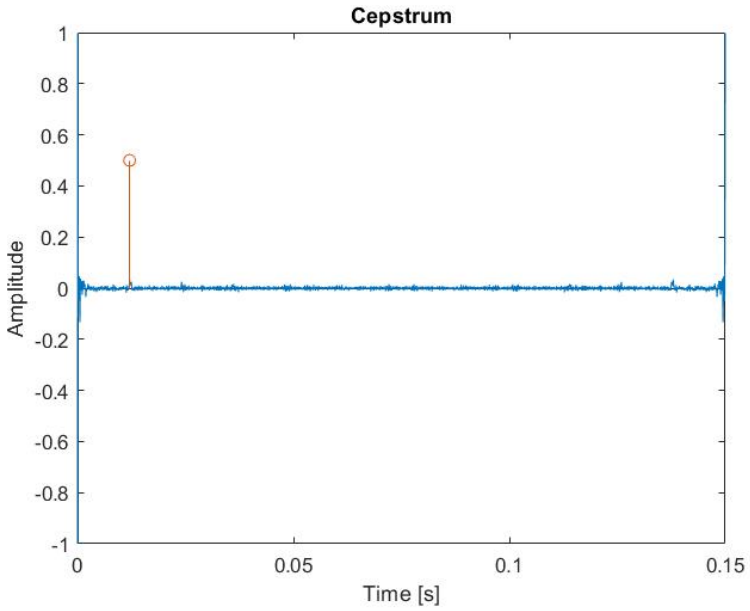


Figure 5.11: The total cepstrum in the time domain of a male speaking the letter 'a'

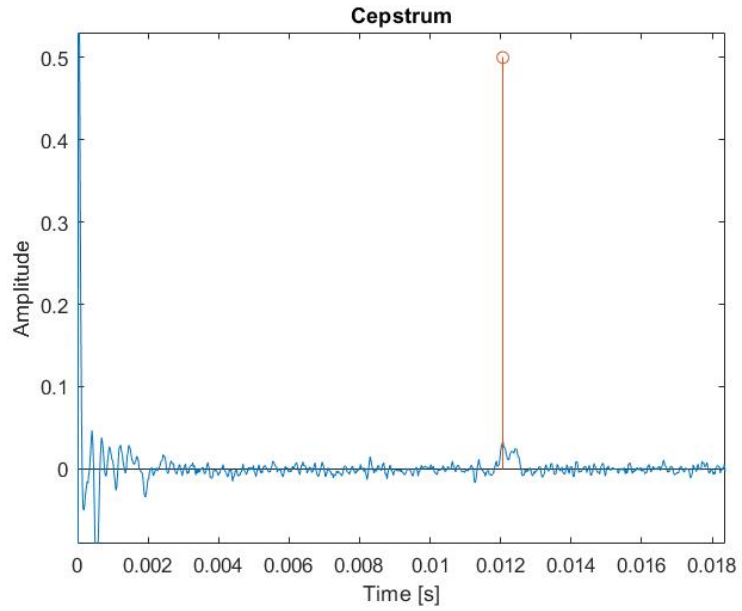


Figure 5.12: The cepstrum zoomed in in the time domain of a male speaking the letter 'a'

5.4.4. F0 tracking

In Figure 5.13 a figure can be seen in which the tracking algorithm is tested for an increasing intonation while speaking the letter 'a'. It can be seen that there are no outliers in the spectrum anymore. In Figure 5.14 it can be seen that the tracker function ignores some jumps because of the fact that the jump was too high, but it jumps after 3 times.

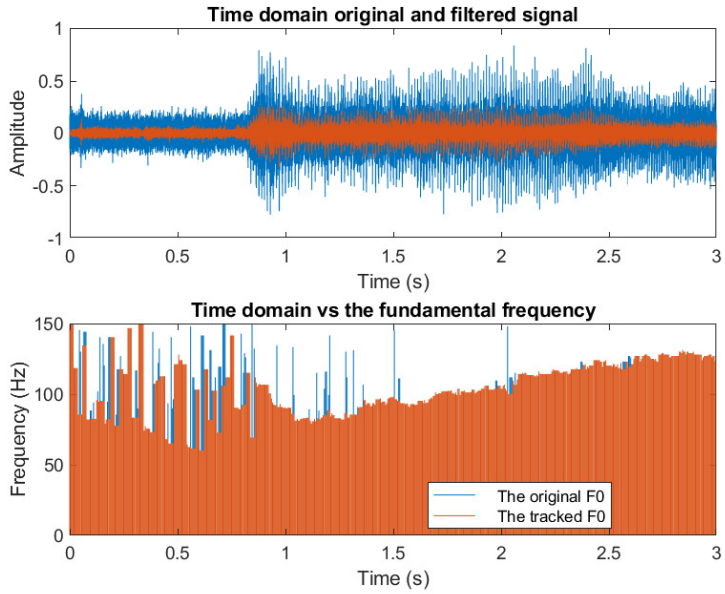


Figure 5.13: F0 determination algorithm example with multiple fundamental frequencies, measured at a sample frequency of 16 kHz while increasing the intonation of the letter 'a'

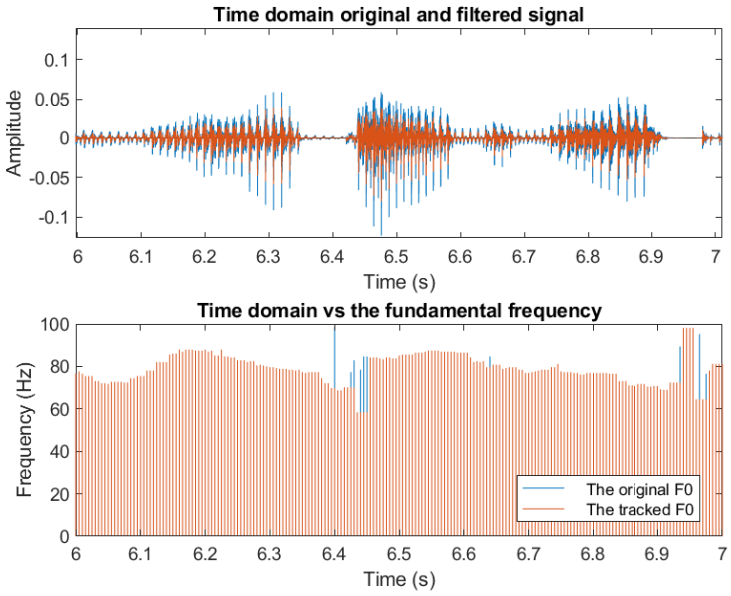


Figure 5.14: F0 determination algorithm example with multiple fundamental frequencies, measured at a sample frequency of 16 kHz while speaking the letter 'a' on different tones

5.4.5. Bandpass filter

The original bandpass filter needed to be tested as well. A figure of the original bandpass filter can be seen in Figure 5.17 It can be seen that the filter is a hard rectangular filter from 0 to 500 Hz and from 4000 Hz to the end of the spectrum.

5.4.6. Whole system

The whole system is tested for a different set of values and the I is plotted vs the filter range for different window lengths. This plot can be seen in figure Figure 5.6. In this figure it can be seen that the best result is realised with a window length of 32 and a frequency range does not really matter anymore at that point. It

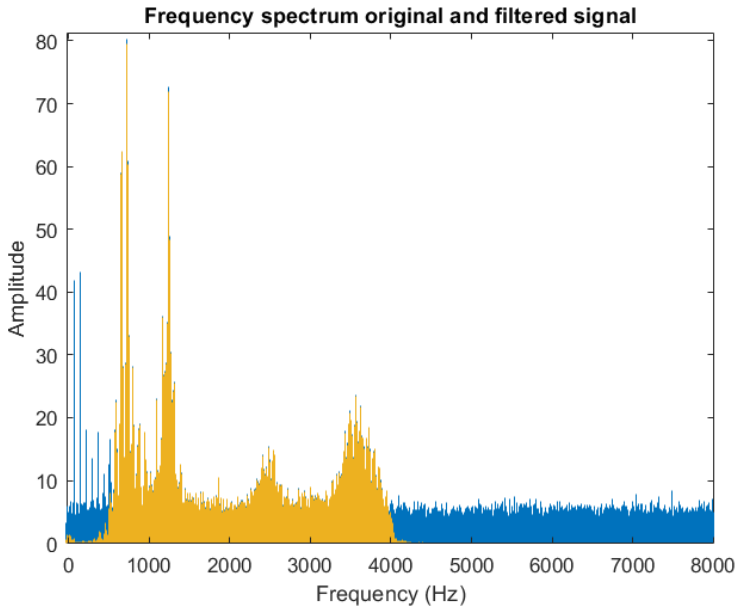


Figure 5.15: The frequency spectrum of the original signal with noise and the filtered signal using a simple, rectangular window, bandpass filter. The passband is 500Hz to 4kHz.

should be taken into consideration that some results may result in no change at all. This means that if the frequency range is equal to 30 Hz, the filter will do nothing anymore.

The filter is tested in another way, by use of a frequency spectrum. In Figure 5.16 the original signal without noise can be seen. In Figure 5.17 the original signal with added noise can be seen with its filtered signal next to it. It can be seen that the filtered signal follows the original signal, but not exactly reconstructs it. This indicated that there will still be some noise in the audio signal, even though it is way less.

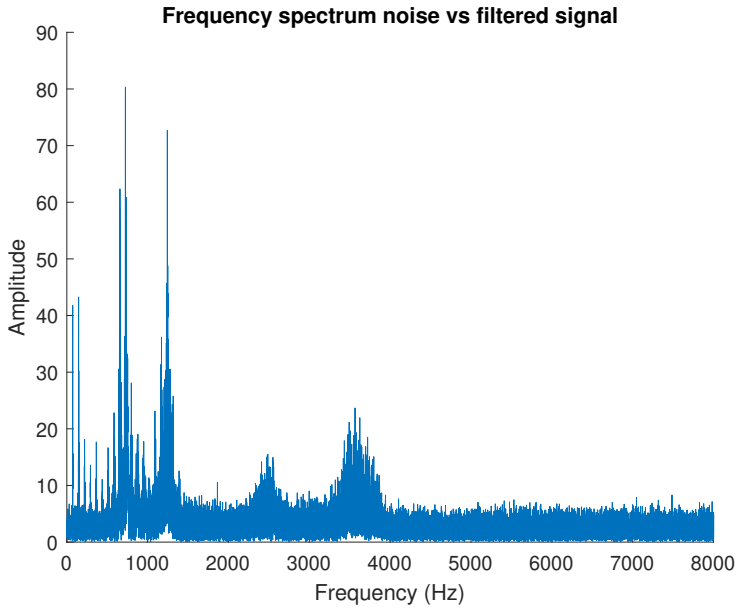


Figure 5.16: The original frequency spectrum

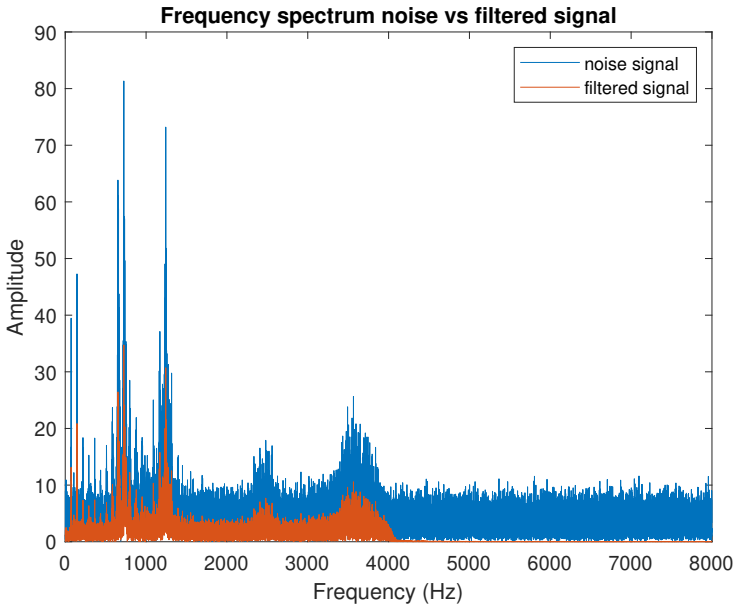


Figure 5.17: The frequency spectrum of the original signal with noise and the filtered signal using the F0 dependent filter

6 | Circuit design

To connect the microphone with an analogue output to the DSP first the data needs to be quantified. This job will be done by an analogue to digital converter (ADC). Before the ADC can quantize the signal the signal will need to be strong enough. Also the signal can not contain any frequency information above half the Nyquist frequency.

6.1. Sample frequency Fs

By quantising the analogue signal the data is translated from the analogue to the digital domain. According to Nyquist all frequency information below half the sampling frequency Fs can be reconstructed after sampling. When quantising, all frequency information above Fs/2 will be mapped onto the lower frequencies. So data from higher frequencies appear at lower frequencies. This phenomenon is called aliasing. As aliasing distorts the signal of interest it should be prevented. This can be done by simply applying a low pass filter before the quantising circuit element, otherwise called the anti aliasing filter. For the anti aliasing filter design several essential frequencies are calculated. Starting off with the sampling frequency Fs, which is constrained by the maximum frequencies of interest in the speech signal which was determined to be around 4kHz section 1.1. Although Fs does not influence the frequency resolution for fft, it does determine the data sizes to be computed. In Table 6.1 it can be seen that the intelligibility does not depend on Fs, the table also shows that the intelligibility does not change if the system is filtered at 4kHz instead of 8kHz. Therefore, Fs can be chosen at 16 kHz. At $Fs = BW * 2 = 16$kHz all frequency information of interest can be retrieved and some room is left for the filter transition band. So Nyquist is satisfied as $Fs/2 > 4$kHz and this passband comprises the speech spectrum of interest with enough margin but still keeps the sample sizes limited to later ensure fast computing. Fs is thus chosen as low as possible as to save computational time to minimise latency and a higher Fs would not add any value in terms of intelligibility. The low Fs does however imply a rather strict anti aliasing filter is required.

Fs [kHz] & filter frequency range (frng) [kHz]	I [b/s]
Fs = 16 & frng = 4	529.4779
Fs = 16 & frng = 8	534.0706
Fs = 44.1 & frng = 4	572.7099
Fs = 44.1 & frng = 8	529.4915

Table 6.1: The results of the SIIB algorithm on different sample rates and filter frequency ranges.

6.2. Anti aliasing filter

After the sample frequency is calculated, the passband of the anti aliasing filter can be determined. The spectrum of interest of speech is determined to be up to 4kHz, so $f_{pass} = 4$kHz. The Stop band start frequency can be calculated as the maximum frequency in the signal allowed to avoid aliasing, which is $f_{stop} = Fs/2 = 16/2 = 8$kHz. The transition band is then $f_{trans} = f_{stop} - f_{pass} = 4$kHz. These bands are depicted in Figure 6.1.

How much the data in the stop band should be attenuated is decided by the desired signal to noise ratio (SNR). Which should be optimised but the filter order will grow with a higher SNR as the rolloff will need to be steeper for a higher SNR. Therefore a SNR of 24dB is chosen. As f_{trans} spans about $log_{10}(f_{stop}/f_{pass}) \approx 0.3$ of a decade, the roll off needs to be about 80dB/decade. So a 4th order filter is required.

The filter type is chosen to be a Butterworth filter. Though a Butterworth filter has a more slow roll off, it is chosen for its linear phase behaviour and its relatively low impact on the passband data. First the normalised poles $S(k)$ for a normalised 4th order Butterworth filter are calculated using Equation 6.1 from [1]. It is a generalised equation calculating the poles $S(1:N)$ and the mirrored poles $S(N:k)$ for an Nth order Butterworth filter by its definition. This way it is guaranteed the designed filter is stable as all normalised poles are on the unit circle in the left half plane.

$$\exp^{i\frac{(2*k-1+N)*\pi}{2*N}} \text{ with} k = 1,2,...,2*N \tag{6.1}$$

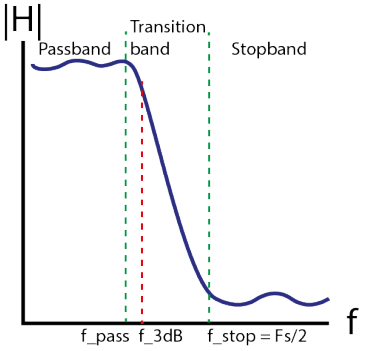


Figure 6.1: Indicative figure envisioning the passband, transition band and stopband for a low pass filter. Also some relevant parameters are depicted.

After the poles are calculated, these can be used to find the RC ratios corresponding to each pole. Typical values for R and C are selected aiming to get R in the kΩ range and C in the nF range. The next step is to denormalise the found poles so the cutoff frequency is shifted to the desired place between f_{pass} and f_{stop}. More on this is discussed in section 7.2.

6.3. ADC

After the anti-aliasing filter enures that there is no frequency information anymore after 8000 kHz in the signal data, the analogue signal needs to be converted to bits to make sure the DSP can work with the input data.

specification

To make sure the ADC works with the DSP it is important that the ADC uses I^2S protocol, since the DSP uses this protocol as well. The ADC needs to make sure that the speech still has an SNR which can be considered to be clean speech. The ADC adds quantization noise, so this should be taken into consideration. The quantization noise is the maximum amount of inaccuracy due to a constraint in number of bits. The quantization noise can be determined by use of Equation 6.2. With the result of Equation 6.2 the SNR of the speech can be determined by use of Equation 6.3. With a combination of these equations and the knowledge that speech is considered clean if the SNR is 30 [4] dB or higher, the minimum number of bits needed can be determined. If the equations are rewritten, the equations result in Equation 6.4. With this equation it is determined that the number of bits needed for an SNR of 30 dB is equal to 5. The closest number of bits for the ADC is 8. This is the reason why the minimum number of bits of 8 is chosen.

$$q_N = \frac{V_{supply}}{2^b} \tag{6.2}$$

$$SNR = \frac{V_{supply}}{q_N} \tag{6.3}$$

$$b = log_2(10^{SNR/20}) \tag{6.4}$$

The equation parameters are described below:

- q_N: the quantization noise in V
- V_{supply}: the supply voltage
- b: the number of bits
- SNR: the signal to noise ratio

Selection

In order to be sure to make the correct decision about which ADC is going to be used, a few ADC's have been put into an choice system as can be seen in Figure B.3. The selected ADC's are rated on some characteristics. These characteristics are: the supply voltage, the power dissipation, the number of bits it converts the audio

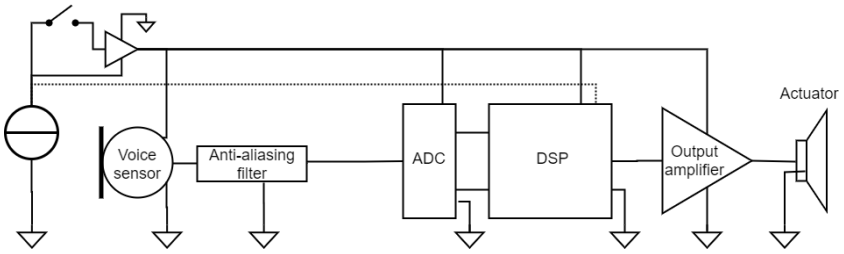


Figure 6.2: Simplified circuit for the power supply of the system, showing the switch to be placed near the stoma, the buffer and the main power connections to the components of the system. Also the low power supply to the DSP is shown with the dashed line.

signal to, the sample frequency, the clockrate, the size, the costs and the rise time. The selected ADC has the highest score. This is the TLV320ADC3101-Q1. This ADC has small dimensions, low power and uses more bits then the calculated minimum amount of bits. Its data sheet can be found at [22].

6.4. Power supply and switch

For the power supply a battery was chosen. In this section it is connected to the system components. The microphone, ADC, DSP and amplifier all need direct power from the battery. The anti aliasing filter and actuator are passive components. The microphone and ADC can be directly connected to the battery. More on connecting the DSP and amplifier in the shift and output sister thesis, [11].
The microphone uses the power supply for a simple buffer. The maximum power dissipation is $3.7 * 220 = 0.814$mW, calculated from the maximum supply voltage and maximum supply current from the data sheet, [14]. The ADC uses 10mW at mono 48kHz as is stated in its datasheet, [22]. The DSP static power consumption when not calculating is 1.12 mW. And the DSP active power consumption is estimated at 51.9mW at 100MHz. For the amplifier, two concepts are worked out. A Piezo amplifier using 1.66W and a class D mono amplifier using 2W. The power consumption from the DSP and the amplifiers are taken from the shift and output thesis [11].
The total power consumption is then only about 1.12mW at standby and when active at most $0.814+10+51.9+2000 = 2062.7$mW. Using Equation 4.1, a battery size of $C_{Bat} = \frac{((2.0627 \cdot 0.1928)+0.00112) \cdot 2 \cdot 1000}{3.7} \cdot 1.2 = 258.687$mAh is necessary.
The peak current drawn can be found as $2.0627/3.7 = 0.5565$A. So at 556.5mA the peak current is safely below the maximum peak current 980mA.
Then the switch has to be integrated. The microphone, ADC and amplifier do not suffer from startup delay. These components can be directly be cut off from power to save energy when the user is not speaking. The DSP however needs a low power pin as is discussed in the shift and output thesis [11].
The switch was chosen to be a physical pressure switch placed near the stoma for easy and natural turning on the system. It is not desirable to let all system power flow from the battery to the switch back to the system. So a buffer is placed between the switch near the stoma and the V_{dd} of the system. Now the switch works as a sensor and the actual power cutoff and power supply happens in the devise casing. An impression of the power connection circuit is given in Figure 6.2.

6.5. Size

To determine the physical dimensions of the design. All components area's were added together and multiplied by an imperfection margin. This imperfection margin takes note in the fact that all the components will not be placed perfectly on each other. The resulting size is 31.10 cm^3.

7 | Discussion

The goal of the thesis was to improve speech for LP. Even though the methods suggested in this paper do work, it is not near the perfect outcome which is desired. Therefore there are a lot of improvements which can be done.

7.1. Filter

The filter did what it is supposed to do, but it does not do a perfect job. This means that the filter does reduce the noise in the signal which results in a better audio signal, but there still is noise in the audio signal. It has been shown that the filter as a whole works the best if the window length is set equal to 32 and the filter range can be a variety of choices. The filter has proven to work for multiple vowels.

7.2. Hardware

The microphone which is selected looks promising, but it needs to be build into a casing which is not designed yet. Also real life testing was not possible, so before the system can be called a success naturally tests will need to be conducted.
The battery seems have sufficient capacity. However the discharge behaviour should be tested and the effects of the relative high typical current should be examined before it can be concluded the optimal battery is selected.
For the anti aliasing filter significant future research is required. After the poles are found for a normalised cutoff frequency and the RC values are scaled to realistic values as explained in section 6.2, the capacitance is denormalised by dividing the capacitance with the radial cutoff frequency as given by Equation 7.1. Now the cutoff frequency has shifted to the desired value and the filter parameters are known.

$$C = \frac{C_{normalised}}{2 * \pi * f_{3dB}} \tag{7.1}$$

The RC 4th order Butterworth filter is then simulated using the found R and C values. The simulation is done in LTspice to verify the functioning. The result is shown in Figure 7.1.

The damping at f_{stop} is sufficient with the designed filter. The cutoff frequency however is not close to 4kHz. This is very problematic and solutions like a higher Fs so the anti aliasing filer needs to be less complex or another filter type need to be considered.
Although the power system and switch are a relatively simple part of the design, no switch was selected or buffer circuit was designed or selected.

Total simulation: The hardware implementation has never been tested and it thus can not be concluded to work properly. The chosen devices do meet the requirements which they needed to.

7.3. Meeting of the requirements

Requirements were set in **??**, most of these requirements are followed up as will be discussed in the next sections. First the general requirements.

General requirements

The system comprises a microphone, signal processing part and an energy source. Although no prototype was build TESSA is designed to be worn on the body, all selected components are small and the estimated size of the total system is 31.10 cm^3. The input bandwidth is 100Hz to 4kHz, which does fall short of the 50Hz lower limit from the SOR, but this only the heat production was not measured concluding the general requirements.
This means that the system (if physically made) will be able to be wearable on the body, the input bandwidth is between 50 Hz and 16 kHz. The microphone is chosen to be something other than a headset. The system provides 2 hours of speaking time. The device is rechargeable. The charging time is less then 8 hours. The goal which is not measured is: The temperature stays below 36 degrees. Next to guidelines, this thesis also had a main goal. The device should be able to make the LP sound more natural and more intelligible. This goal was

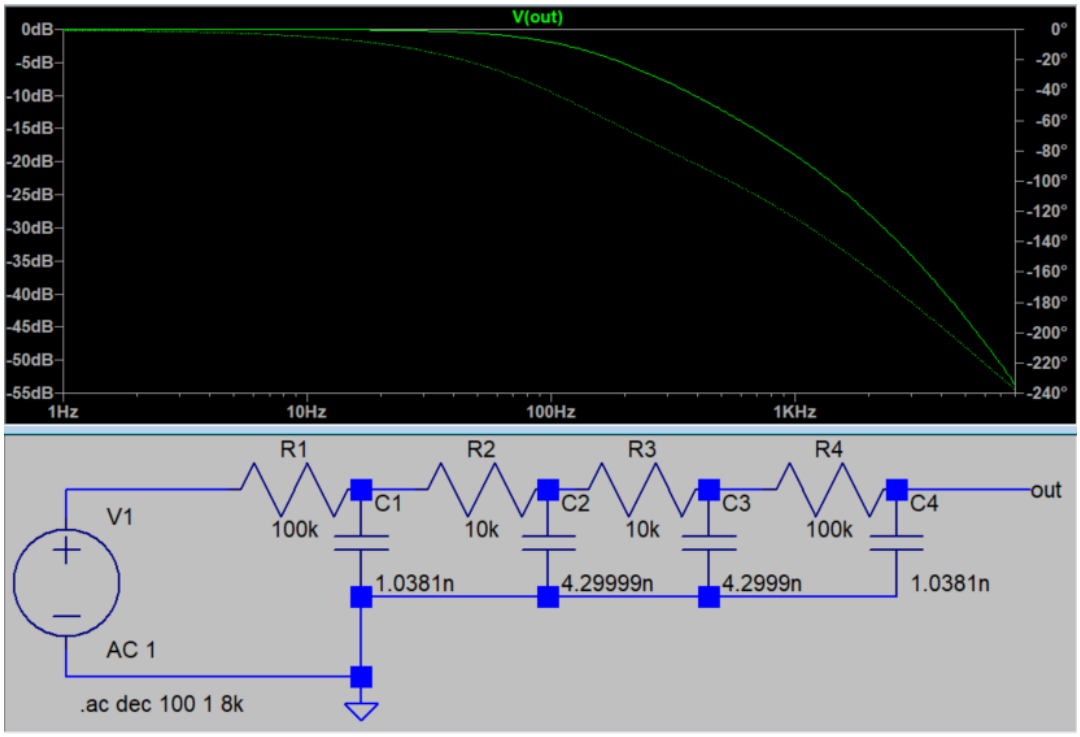


Figure 7.1: The anti aliasing circuit implementation with component values and simulation. add the indicated -3dB freq, roll off, stop band attenuation, passband attenuation.

realised, but the product did not do a perfect job. This means that the way in which LP would speak while using our device would definitely improve but not solve their speaking deficiencies. In this thesis seven conclusions were to be drawn. First off, the best microphone to be used is the MEMS electret SPU0414HR5H-SB microphone. The battery which should be used is the DRJ3555. The ADC to be used is the TLV320ADC3101-Q1. The sampling frequency should be 16 kHz. A correct way of filtering is concluded to be a fundamental frequency dependent filter, resulting in the need of a fundamental frequency determination function which was concluded to be the cepstrum. A windowing technique was used for which the length was determined to do best if it was set to 32. The best working windowing technique was determined to be the hann window.

7.4. Recommendations for future work

The work which has already been done can be improved in the future. Some recommendations are mentioned.

Recommended future work filter

A way to improving the filter as it has already been made at this moment is by making use of a time difference of arrivel (TDOA) function. This function will measure the difference in time between two audio signals, measured from 2 microphones. If this difference in time is not equal to a pre determined time-difference, the function will flag the audio as noise. This will result in the fact that other people's speech will not be caught and processed.
The algorithm can be improved as well by use of a peakfinder. A peakfinder was already written for this program and it can be used. The problem with the peakfinder is the fact that it did not change the fundamental frequency because of all the other accuracy measures which were already taken.
In this paper the methods of improving the intelligibility are by using a filter. In order to make a device which improves the LP speech in a way better manner than that has already been provided within this thesis, research needs to be done to speech reconstruction. Speech consists of different characteristics and it can be synthesized, as is done in phone calls. If it is possible to determine the characteristics of an LP voice before they loose their voice, then it is possible to reconstruct their voice by use of the characteristics.

Recommended future work hardware design

Furthermore in the future the proof of concept could be made in real life, meaning that the designed device needs to actually be simulated/build and then tested. In this manner the actual imperfections of the component can be studied and resolved.
The device at this point is not made to be waterproof, or tested to be wearable, these are future implementations as well. The coating thus still needs to be designed. This coating will also have to take into account that heat that the device makes, meaning that the device cannot get above the comfortable wearing temperature.
At this moment the device also still needs the user to push the throat with his finger in order to be able to speak. In further research, this should be made automatically is well. This could be done by use of a pressure sensor. For instance the user could make use of a pressure bump, resulting in the fact that the sensor measures the will to speak. Then the stoma could be closed automatically and the user will be able to speak.
Further research into the ADC is also needed. The ADC is determined to be accurate up untill 13mW, which was determined to have an SNR of 48 dB. 48 dB is way above the accuracy needed for clean speech (which is 24 dB) therefore a ADC with 8 bits is enough, and an ADC with 4 bits is too close to the edge. The ADC was not chosen perfectly for I2s, which is an problem. It should however do what it is supposed to do. Due to time constraints, it was not possible to simulate everything accordingly.
Take note in the fact that due to the corona virus, this paper is mostly made theoretical, resulting in the assumption that every component works accordingly.

8 | Conclusion

The goal of this thesis is to design a device which makes the LP sound more natural and more intelligible. This goal is approached, but not met. The requirements which were set in the beginning of the process in communication with the client were all met but one. The requirements which were met or met within margin are: the bandwidth is sufficient as it is 100 Hz to 4 kHz, the system is wearable on the body as the chosen components are all limited in dimensions. And the system provides 2 hours of speaking time. Furthermore the operating temperature of the system cannot really be determined, since this will depend on the isolation of the components by the casing. Which is a matter which is out of the scope of this project due to the corona virus limiting us to a theoretical design.
All the individual parts of the filter algorithm were tested and work up to desired level. The filter as a whole works, if the intelligibility is improved is however a very subjective matter.
For the microphone a MEMS electret microphone is selected. For the power source a rechargeable Li-ion button cell battery with a capacity of 500 mAh is selected. For the ADC a low-power 16 bit ADC which works on the I^2S protocol is selected.
Finally the goal is approached as an innovative system is designed which is proven to edit the recorded speech data without deteriorating the quality. The system is however not in a readily usable state, a lot of further research needs to be conducted before the system can be implemented. Also the naturallity and intelligibility of a voice stays a subjective matter, so it is for it can not be concluded this goal is reached.

Bibliography

[1] Luis F. Chaparro. *Signals Ans Systems Using Matlab second edition*. 2015. ISBN 978-0-12-394812-0.

[2] Jingdong Chen, Yiteng Huang, and Jacob Benesty. *Filtering Techniques for Noise Reduction and Speech Enhancement*, pages 129–154. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-662-11028-7. doi: 10.1007/978-3-662-11028-7_5. URL https://doi.org/10.1007/978-3-662-11028-7_5.

[3] PH.D. P. DELAERE M.D. PH.D. J. WOUTERS PH.D. P. UWENTS M.D. F. DEBRUYNE, M.D. Acoustic analysis of tracheo-oesophageal versus oesophageal speech. *The Journal of Laryngology and Otology*, 108: 325–328, 1994.

[4] Dave Gelbart. How is the snr of a speech example defined, 2000. URL https://www1.icsi.berkeley.edu/Speech/faq/speechSNR.html#:~:text=A%20local%20SNR%20of%2030dB,and%20noise%20energy%20the%20same).

[5] Bolognone R.K. Graville D.J., Palmer A.D. Voice restoration with the tracheoesophageal voice prosthesis: The current state of the art. *Doyle P. (eds) Clinical Care and Rehabilitation in Head and Neck Cancer.*, 2019.

[6] Zdeněk Hanzlíček, Jan Romportl, and Jindřich Matoušek. Voice conservation: Towards creating a speech-aid system for total laryngectomees. pages 203–212, 2013. doi: 10.1007/978-3-642-34422-0_14. URL https://doi.org/10.1007/978-3-642-34422-0_14.

[7] M. HASAN and Tetsuya Shimamura. A fundamental frequency extraction method based on windowless and normalized autocorrelation functions. pages pp.305–309, 01 2012.

[8] F Mertens M De Bodt P.H Van de Heyning Heylen, F.L Wuyts. Normative voice range profiles of male and female professional voice users. *Journal of Voice*, 16:1 – 7, 2002. ISSN 0892-1997. doi: https://doi.org/10.1016/S0892-1997(02)00065-6. URL http://www.sciencedirect.com/science/article/pii/S0892199702000656.

[9] Shannon Hilbert. Fft zero padding, 2013. URL https://www.bitweenie.com/listings/fft-zero-padding/.

[10] National instruments. Understanding ffts and windowing, 2018. URL https://circuitdigest.com/tutorial/classes-of-power-amplifier-explained.

[11] B. Fennema J. Breukelen van, T. Alers. Bachelor afstudeer project groep I, TESSA shift and output. 2020.

[12] J.W. Tukey J.W. Cooley. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, pages 19 – 279, 1965.

[13] A. Keltz. Latency and digital audio systems. URL http://whirlwindusa.com/support/tech-articles/opening-pandoras-box/.

[14] *Amplified SiSonic Microphone*. Knowles, 2012.

[15] Lund S.P. Persson R. et al. Kristiansen, J. A study of classroom acoustics and school teachers' noise exposure, voice load and speaking time during teaching, and the effects on vocal and mental fatigue development. *Int Arch Occup Environ Health*, 87:851 —- 860, 2014. doi: https://doi.org/10.1007/s00420-014-0927-8. URL https://link.springer.com/article/10.1007/s00420-014-0927-8#Tab1.

[16] Trialsight Medical Media. Anatomy of the larynx, 2008. URL http://www.hawaiivoiceandspeechstudio.com/blog/archives/05-2016.

[17] Robert B. Randall. A history of cepstrum analysis and its application to mechanical problems. *Mechanical Systems and Signal Processing*, 97:3 – 19, 2017. ISSN 0888-3270. doi: https://doi.org/10.1016/j.ymssp.2016.12.026. URL http://www.sciencedirect.com/science/article/pii/S0888327016305556. Special Issue on Surveillance.

[18] *Rechargeable LI-ION Batteries*. RJD, 2019.

[19] R.C. Hendriks S. Van Kuyk, W.B. Kleijn. An evaluation of intrusive instrumental intelligibility metrics. 2017.

[20] Ljiljana Sirić, Dario Sos, Marinela Rosso, and Sinisa Stevanović. Objective assessment of tracheoesophageal and esophageal speech using acoustic analysis of voice. *Collegium antropologicum*, 36 Suppl 2:111–4, 11 2012.

[21] Morten Stove. Facts about speech intelligibility, 2016. URL https://www.dpamicrophones.com/mic-university/facts-about-speech-intelligibility.

[22] *Low-power Stereo ADC With Embedded miniDSP for Wireless Handsets and Portable Audio*. Texas Instruments, 2012.

[23] Hunter E. J. Svec J. G. Titze, I. R. Voicing and silence periods in daily and weekly vocalizations of teachers. *The Journal of the Acoustical Society of America*, 121:469 —- 478, 2007. doi: https://doi.org/10.1121/1.2390676. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6371399/.

[24] K. Yu and S. Young. Continuous f0 modeling for hmm based statistical parametric speech synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1071–1079, 2011.

[25] Zhaoyan Zhang. Mechanics of human voice production and control. *The Journal of the Acoustical Society of America*, 140(4):2614–2635, 2016. doi: 10.1121/1.4964509. URL https://doi.org/10.1121/1.4964509.

A | Morphological maps

All morphological maps containing the choice process for the optimal concept are shown below. For the sensor as an explanation all steps are shown. First all possible concepts are gathered by dividing concepts in subjects. Then for all subjects of the concepts different options are considered. So every morphological map entry is a sub solution of a concept. A concept is then composed by selecting different sub solutions. A typical morphological map is shown for the sensor A.1. To select the optimal sub solutions for the optimal concept, all morphological map entries are rated for relevant categories as shown in A.2. For the switch the weight factors for the categories are calculated by choosing which category is more important then the other to find the weight factors as objectively as possible as shown in A.7. The morphological map entries with the highest score for each subject are then selected. In the figures these sub solutions are outlined in red. From the outlined sub solutions the final concept arises. The result is the optimal concept as depicted in A.3.

Subjects	Options						
Location	Mouth	Neck	Mouth and Throat	Ear	Forehead (glasses)	Forehead (hair)	
Fixation	Sticky on	Earpiece	Headset	Clip on			
Number of microphones	1	2	3				
Directionality	Omnidirectional	Unidirectional	Cadioid	Bidirectional			
Technology	Moving coil	Condenser	Electret	Ribbon	Crystal / ceramic	Boundary / PZM	Carbon
Data Transport	Analogue	PWM	PDM	I2S	I2C		

Figure A.1: Morphological map of the sensor to envision all considered concepts by illustrating the sub solutions of every concept

Map entries:	BW	Size	Weight	Sensitivity	Price	SNR	Noticability	Flexibility	End score	
Mouth	1	0	0	1	0	1	-1	1	4	Mouth
Neck	-1	0	0	-1	0	1	1	1	4	Neck
Mouth and Throat	1	0	0	1	0	1	-1	1	4	Mouth and Throat
Ear	0	0	0	0	0	-1	1	1	4	Ear
Forehead (glasses)	1	0	0	0	0	0	1	0	3	Forehead (glasses)
Forehead (hair)	1	0	0	0	0	0	1	-1	0	Forehead (hair)
Stick on	0	0	0	-1	0	-1	1	-1	-3	Stick on
Ear piece	1	0	0	0	0	0	0	1	4	Ear piece
Headset	1	-1	0	1	0	1	-1	1	3	Headset
Clip on	-1	0	0	0	1	-1	1	1	4	Clip on
1 mic	0	1	1	0	1	0	1	1	8	1 mic
2 mics	1	1	1	1	0	1	1	1	10	2 mics
3 mics	1	0	0	1	0	1	0	0	3	3 mics
Omniderectional	1	0	0	1	0	0	1	1	7	Omniderectional
Unidirectional	1	0	0	1	0	1	0	-1	0	Unidirectional
Cardoid	1	0	0	1	0	1	0	-1	0	Cardoid
Bidirectional	1	0	0	0	0	0	0	-1	-2	Bidirectional
Moving coil	1	-1	-1	-1	1	0	0	1	2	Moving coil
Condenser	1	-1	-1	1	0	0	-1	-1	-5	Condenser
Electret	1	1	1	1	1	0	1	1	10	Electret
Ribbon	1	0	0	1	0	0	0	0	2	Ribbon
Crystal / ceramic	1	1	1	1	0	0	1	0	6	Crystal / ceramic
Boundary / PZM	1	-1	-1	0	0	0	-1	-1	-6	Boundary / PZM
Carbon	0	-1	-1	-1	0	-1	-1	1	-3	Carbon
Analogue	0	0	0	0	0	-1	0	1	2	Analogue
PWM	0	0	0	0	0	0	0	0	0	PWM
PDM	0	0	0	0	0	0	0	0	0	PDM
I2S	0	0	0	0	0	0	0	0	0	I2S
I2C	0	0	0	0	0	0	0	0	0	I2C

Legend:	Category	Weight	Description
Positive	BW	1	Achievable Bandwidth
Negative	Size	1	Size of transducer system
Neutral	Weight	1	Does the trasducer produce a measurable voltage or current
	Sensitivity	1	Achievable sensitivity of transducer
	Price	1	General price of technology
	SNR	1	Signal to noise ratio
	Noticability	2	Is the transducer very visible or even eye catcing
	Flexibility	3	Flexibility of use, e.g. can it be worn under clothes and does it catch wind, is it easily put in place

Figure A.2: Rating of sub solutions of the morphological map for the sensor to efficiently choose the optimal concept, all winning sub solutions for each subject are outlined in red.

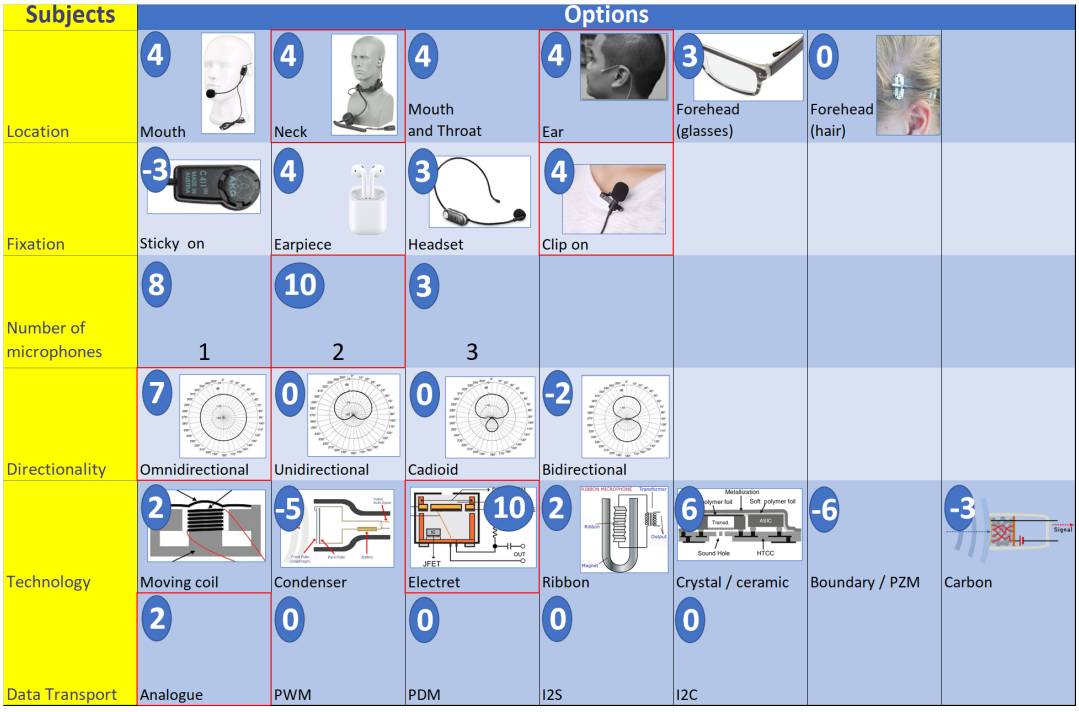


Figure A.3: Morphological map for the sensor with indicated final scores of sub solutions and in red outlined the chosen sub solutions composing the chosen concept

Figure A.4: Morphological map for the power source with indicated final scores of sub solutions and in red outlined the chosen sub solutions composing the chosen concept

Map entries:	Charge time	Size	Flexibility of use	Cost	Capacity per volume	Lifetime	Total score	
Rechargable	0	0	1	0	1	0	5	Rechargable
Replacable	1	0	0	0	0	1	3	Replacable
Rechargable & replacable	1	0	1	0	0	1	5	Rechargable & replacable
Li-ion	1	1	0	0	1	1	9	Li-ion
Li-polymer	1	0	0	0	0	0	1	Li-polymer
NiMH	-1	0	0	1	-1	0	-3	NiMH
Chemical	0	1	0	0	0	0	3	Chemical
Flywheel	1	-1	-1	-1	1	1	0	Flywheel
Hydrogen	-1	-1	-1	-1	-1	1	-8	Hydrogen
Wireless	0	0	1	-1	0	0	1	Wireless
Micro USB	1	0	1	1	0	-1	2	Micro USB
USB C	1	0	1	-1	0	0	2	USB C
simple connector	1	-1	0	1	0	0	-1	simple connector
external charger	1	1	0	-1	0	0	3	external charger

Legend:	Category:	Weight:	Description:
Positive	Charge time	1	Time to fully recharge power carrier
Negative	Size	3	Dimensions of power carrier
Neutral	Flexibility of use	2	Ability of power carrier to be moved around on the human body
	Cost	1	Cost of power carrier
	Capacity per volume	3	Amount of energy strored per unit volume
	Lifetime	2	How many charge cycles does the power carrier allow before deteriorating

Figure A.5: Rating of sub solutions of the morphological map for the power source to efficiently choose the optimal concept, all winning sub solutions for each subject are outlined in red.

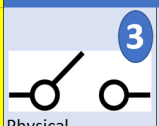
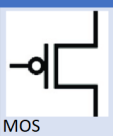
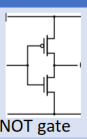
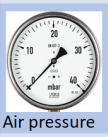
Subjects	Options					
Switch technology	 3 Physical	 1 MOS	 -3 Software	 1 NOT gate		
Switch method	 6 Button	 7 Air pressure	 3 Sound level	 -3 Rocker	 -3 Wireless	 -4 Capacitive
Location	-5 Chest	5 Throat	5 Integrated			

Figure A.6: Morphological map switch system with indicated final scores of sub solutions

Map entries:	Speed	Effort	Noticability	Naturallity	Power	End score	
Physical	1	0	0	0	1	3	Physical
MOS	1	0	0	0	0	1	MOS
Software	-1	0	0	0	-1	-3	Software
NOT gate	1	0	0	0	0	1	NOT gate
Button	1	0	0	1	1	6	Button
Air pressure	1	1	0	1	0	7	Air pressure
Sound level	-1	1	0	1	-1	3	Sound level
Rocker	1	-1	0	-1	1	-3	Rocker
Wireless	-1	0	0	0	-1	-3	Wireless
Capacitive	1	-1	0	0	-1	-4	Capacitive
Chest	0	-1	1	-1	0	-5	Chest
Throat	0	1	-1	1	0	5	Throat
Integrated	0	1	1	1	-1	5	Integrated

Category	Speed	Effort	Noticability	Naturallity	Power	Deduced weight	Description
Speed	x	0	0	1	0	1	Switch time
Effort	1	x	1	0	1	3	Easyness for the user to use switch
Noticability	1	0	x	0	0	1	How visible is the switch mechanism
Naturallity	0	1	1	x	1	3	Is the switch action natural for the user
Power	1	0	1	0	x	2	Power efficiency of switch and power saving

Figure A.7: Rating of sub solutions of the morphological map for the switch to efficiently choose the optimal concept

B | Component selection rating

B.1. Battery selection rating

Batteries:	Chemical type	Voltage (V)	Capacity (mAh)	Energy (mWh)	Size (mm)	Volume (mm^3)	Enegry per volume (mWh/mm^3)	Cost (euro)	Total score
BrightTea	Li-ion	3,7	250 (1)	925 (0)	31*20*5	3100 (1)	0,298 (0)	€ ? (0)	4
LIR 2450	Li-ion	3,6	120 (0)	432 (-1)	5*24,5	2357(1)	0,183 (-1)	€ 6,15 (1)	-2
IEC-HR03	NiMH	1,2	1000 (1)	1200 (0)	4,30*44,5	6687 (0)	0,179 (-1)	€ 2,15 (1)	1
RDJ3555	Li-ion	3,7	500 (1)	1850 (1)	35,2*5,7	5547 (1)	0,334 (1)	€ 28,00 (0)	8
Endurance 6LR61	NiMH	8,4	200 (0)	1680 (0)	24,5*16	4926 (0)	0,341 (1)	€ 3,79 (1)	1
NP-BX1 X-serie	Li-ion	3,6	1240 (1)	4464 (1)	29,9*9,2*42,7	11745 (-1)	0,38 (1)	€ 35,00 (0)	4
LIR 2477	Li-ion	3,6	180 (0)	648 (-1)	24*7.7	3483 (1)	0,186 (-1)	€ 5,02 (1)	-2
18650 cell	Li-ion	3,7	2600 (1)	9620 (1)	18,4*65	17283 (-2)	0,557 (1)	€ 10,07 (1)	2
LP-402933-IS-3	LiPo	3,7	300 (1)	1110 (0)	3,8*29*34	3746 (1)	0,296 (0)	€ 15,92 (1)	4
varta 56455	Li-ion	3,7	600 (1)	2220 (1)	4,35*34*41,5	6137 (0)	0,362 (1)	€ 23,28 (0)	6
RDJ3048	Li-ion	3,7	300 (1)	1110 (0)	30*4,8	3393 (1)	0,327 (1)	€ 10,00 (1)	5

Category:	Capacity	Energy	Volume	E per volume	Cost	Weight	Description:
Capacity	x	0	1	0	1	2	Capacity of the battery in mAh
Energy	1	x	1	0	1	3	Energy stored in battery in mWh
Volume	0	0	x	1	1	2	Volume calculated with the 'size' dimensions in mm^3
E per volume	1	1	0	x	1	3	Amount of energy strored per unit volume in mWh/mm^3
Cost	0	0	0	0	x	0	Cost of power carrier in euros
Chemical type	x	x	x	x	x	0	Chemicals used to store power (only informative)
Voltage	x	x	x	x	x	0	Output voltage in V (only informative)
Size	x	x	x	x	x	0	Dimensions of power carrier in mm (only informative as volume is weighted)

Figure B.1: Score system for the gathered available batteries within the determined specifications. For all criteria the model specific values are given followed by the rating between brackets. The battery with the highest score has a red outlined score and is chosen to be integrated in the design. And below the battery choice weights determined by comparing the independent categories and relatively decide the importance of a weight. The exclusively informative measures are not weighted.

B.2. Microphone selection rating

Microphone	Frequency band	Size	Sensitivity	Impedance	Cost	Directionality	SNR	Techniek	Signal type	Total score
Pro42	70-14kHz (1)	15.5*36.6*54.3mm (-1)	(0)	100 ohm	112,71 (0)	unidirectional (-1)	65 dB (1)	boundary	Analogue	-2
724-3153	100-10kHz (1)	9.7 diameter * 5mm (1)	-40 dB (1)	2.2kohm	1,215 (1)	noise-canceling (0)	(0)	condenser	Analogue	8
771-7011	20-20kHz (1)	6 diameter*2.7mm (1)	-38 dB (1)	2.2kohm	1,736 (1)	omnidirectional (1)	58 dB (0)	electret	Analogue	10
Shure WH20	200-12kHz (1)	6 diameter*4mm (1)	-67dBV/Pa (0)		€ 170,00 (-1)	unidirectional (0)	(0)	Moving coil	Analogue	1
BL-21785-000	20-10kHz (1)	7.78*5.54mm (1)	-69dB (0)	4kOhm	€ 85,54 (0)	omnidirectional (1)	(0)	ceramic	Analogue	5
ECM-821LT	50-16kHz (1)	12 diam * 18mm (1)	-45dB (1)	680ohm	€ 19,34 (1)	omnidirectional (1)	(0)	electret	Analogue	10
SPU0414HR5H-SB	100-10kHz (1)	3.76*2.95*1.1mm (1)	-22dB (2)	400 ohm	€ 1,52 (1)	omnidirectional (1)	59 dB (0)	mems	Analogue	13
MP23ABS1	(0)	3.5*2.65*0.98 mm (1)	-38 dB (1)		€ 2,09 (1)	omnidirectional (1)	64 dB (1)	mems	Analogue	10
ICST43432	50 - 20kHz (1)	4*3*1 mm (1)	-26 dB (2)		€ 4,44 (1)	omnidirectional (1)	65 dB (1)	mems	I2S	14
INMP621	45 - 20 kHz (1)	4*3*1 mm (1)	-46 dB (1)		€ 3,39 (1)	omnidirectional (1)	65 dB (1)	mems	PDM	11

Legend:	Category:	Weight:	Description:
Positive	Frequency band	1	Bandwidth of flat frequency responce of microphone
Negative	Size	2	Dimensions
Neutral	Sensitivity	3	Sensitivity
	Impedance	0	Output impedance of microphone (only informative)
	Cost	2	Typical price of microphone
	Directionality	2	The ability to record in different directions
	SNR	1	Signal to Noise Ratio
	Techniek	0	Microphone technology used (only informative)
	Signal type	0	Microphone output signal type (only informative)

Figure B.2: Score system for the gathered available microphones within the determined specifications. For all criteria the model specific values are given followed by the rating between brackets.

B.3. ADC selection rating

	Voltage (V)	Power (W)	Bits	Fs (kHz)	F_clk (kHz)	Size (mm)	Cost (euro)	Rise time (ns)	End score
TLV320ADC3100	1	-1	1	1	1	1	0	0	12
TLV320ADC3101-Q1	1	0	1	0	1	1	0	1	17
PCM1870A	1	0	1	0	1	1	-1	0	13
TLV320ADC3001	1	-1	1	1	1	1	0	1	15

	Voltage (V)	Power (W)	Bits	Fs (kHz)	F_clk (kHz)	Size (mm)	Cost (euro)	Rise time (ns)	Weight	Description
Voltage (V)	x	1	0	0	0	1	1	1	4	The voltage of the power supply
Power (W)	0	x	1	1	1	1	1	1	6	The power dissipation
Bits	1	0	x	1	1	1	1	1	6	The number of bits
Fs (kHz)	1	0	0	x	1	1	1	0	4	The sample frequency
F_clk (kHz)	1	0	0	0	x	0	1	0	2	The clock frequency
Size (mm)	0	0	0	0	1	x	1	0	2	The dimensions of the ADC
Cost (euro)	0	0	0	0	0	0	x	1	1	The cost
Rise time (ns)	0	0	0	1	1	1	0	x	3	The wake up time

Figure B.3: Score system for the gathered available ADC's within the determined specifications. For all criteria the model specific values are given followed by the rating between brackets.

C | MatLab scripts

C.1. Battery

C.1.1. Capacity calculation

```
%% battery capacity calculator
% Johan Meyer, Folkert de Ronde
% 19-06-2020
% This function determines how much capacity is needed in the battery
Power = 1.5;  % watts
Voltage_battery = 3; % voltage
P_amplifier = 0.8; % efficiency
P_DSP = 0.8;
avg_volume_fac_thuiszitter = (10^8.9*0.19+10^8.25*0.42+10^7.7*0.33+10^7.1*0.038+10^6.8*0.013)/10^9.1; %the average volume factor (waardes gehaald uit een studie)
avg_volume_fac_beweger = 0.7;
battery_safety_margin = 1.2;
time_speaking = 2;

mAh_thuiszitter = ((((Power + P_amplifier)*avg_volume_fac_thuiszitter)+P_DSP)*time_speaking*1000*safety_margin)/Voltage_battery;
mAh_beweger = ((((Power + P_amplifier)*avg_volume_fac_beweger)+P_DSP)*time_speaking*1000*safety_margin)/Voltage_battery;
```

C.2. Filter

C.2.1. Amplitude filter

```
function [buffer_iteration] = A_filter(treshold,stepsize,N,buffer_iteration)
%% Amplitude filter
%Johan Meyer, Folkert de Ronde.
%05-06-2020

%% The input
% the power value for which a datapoint should be considered noise (treshold), the stepsize in which it checks multiple samples (stepsize),
% the number of bins (L), the amplitude signal (buffer_iteration).

%% The output
% the modified amplitude data (buffer_iteration), a 1 or 0 to determine if something is noise or not (noise)

% this part makes sure that low amplitude signals get marked as noise and
% set to 0.
    for j = 1:stepsize:N-stepsize % take steps with stepsize and check if the amplitude data is below a certain treshold
        if sum(abs(buffer_iteration(j:j+stepsize,1))) < (stepsize*treshold)
            buffer_iteration(j:j+stepsize,1) = 0;
        end
    end
end
```

C.2.2. F0 dependent filter

```
function [Buffer] = F0_dependent_filter(w_filter,Buffer, F0, L, Fs, passband_div2,k)
%% F0 dependent filter
% Folkert de Ronde, Johan Meyer
% 04-06-2020
% Filter around all harmonics in the frequency domain

%% The input
% The windowed filter (w_filter) frequency
% fragment (Buffer), the fundamental frequency (F0), The number of bins
% (L), the samplerate (Fs) and the range in which it filters
% (filter_range).

%% The output
% The frequency fragment, but modified (Buffer)

%% filter around F0
    % High pass filter
    % Filter from 0 to the left frequency range off the fundamental
    % Frequency
    k = k+1; % correct for the amount of filters to be skipped
    Buffer = Band_stop_filter_window(w_filter,0,F0*k-passband_div2,'hp',L,Fs,Buffer)
        ; % Function call
    %% Filter around harmonics

    while  (F0*k < 4000-F0) % filter around harmonics up to 4000 Hz
        Buffer = Band_stop_filter_window(w_filter,F0*k+passband_div2,F0*(k+1)-
            passband_div2,'bs',L,Fs,Buffer); % Function call
        k = k+1;
    end
        % filter from the last harmonic (before 4000 Hz) until Fs/2 (thus
        % the whole spectrum)
        Buffer = Band_stop_filter_window(w_filter, F0*k+passband_div2,Fs/2,'lp',L,Fs
            ,Buffer); % Function call

end
```

C.2.3. Band stop filter

```
function frequency_data = Band_stop_filter_window(w,begin_freq,end_freq,filter_type,
    L,Fs, frequency_data)
%% Band stop filter using window
% Folkert de Ronde, Johan Meyer
% 18-06-2020
% Here a band stop filter is implemented using a window chosen in the main
% to avoid spreading of signal power in the time domain due to the effect
% of rectangular windows is the frequency domain.
%% The input
% the window type (w), the frequency at which the band stop filter must
% begin and end (begin_freq, end_freq), The number of bins (L),
% the samplerate (Fs) and the frequency data (frequency_data)
% Also the desired filter window can be chosen as a 'bs'= band stop filter,
% an 'lp'= low pass filter or a 'hp'= high pass filter.


```

```
%% The output
% the modified frequency data (frequency_data)

% choose window in main function:
% %% Window choice filter
% window_choise = 'hamming'; % 'hamming' 'hanning' 'rectangular'
% w_filter_length = 64;
% [unused, w_filter] = window_function(w_filter_length,window_choise); % call function

startsample = ceil(begin_freq*(L-1)/Fs+1); % convert low pass frequency to start sample
endsample = ceil(end_freq*(L-1)/Fs+1); % convert high pass frequency to end sample
startsample_mirrored = (L-1) - endsample; % calculate mirrored sample positions for the mirrored spectrum
endsample_mirrored = (L-1) - startsample; % calculate mirrored sample positions for the mirrored spectrum

w_length = length(w);
windowed_bins_amount = (endsample-startsample)+1; % determine window length by finding
if filter_type == 'bs'  % Make a band-stop filter
    if w_length>windowed_bins_amount
        [~,stop_band] = window_function(windowed_bins_amount,'hanning');  % Window function call
        stop_band = [stop_band(ceil(windowed_bins_amount/2)+1:windowed_bins_amount) ; stop_band(1:ceil(windowed_bins_amount/2))]; % Determine the values for the stopband
    elseif windowed_bins_amount < 3
        stop_band = zeros(windowed_bins_amount,1); % Make sure the function also works when the windowed bins amount is less than 3
    else
        stop_band = [w((w_length/2)+1:w_length) ; zeros(windowed_bins_amount-w_length,1) ; w(1:(w_length/2))]; % create filter window by contagenating the window parts and zeros
    end
    frequency_data(startsample:endsample)=frequency_data(startsample:endsample).*stop_band; % filter using window
    frequency_data(startsample_mirrored:endsample_mirrored)=frequency_data(startsample_mirrored:endsample_mirrored).*stop_band; % filter mirrored spectrum using window

elseif filter_type == 'lp'  % Make a low pass filter
    if w_length>windowed_bins_amount
        [~,stop_band] = window_function(windowed_bins_amount,'hanning'); % Window function call
        stop_band = [stop_band(ceil(windowed_bins_amount/2)+1:windowed_bins_amount) ; zeros(floor(windowed_bins_amount/2),1)]; % Set the stop band values
    elseif windowed_bins_amount < 3
        stop_band = zeros(windowed_bins_amount,1); % Make sure the function still works when the windowed bins amount is less than 3
    else
        stop_band = [w((w_length/2)+1:w_length) ; zeros(windowed_bins_amount-(w_length/2),1)]; % create filter window by contagenating the window parts and zeros
    end
```

```
elseif filter_type == 'hp' % Make a high pass filter
    if w_length>windowed_bins_amount
        [~,stop_band] = window_function(windowed_bins_amount,'hanning'); % Window function call
        stop_band = [zeros(floor(windowed_bins_amount/2),1) ; stop_band(1:ceil(windowed_bins_amount/2))]; % Set the stop band values
    elseif windowed_bins_amount < 3
        stop_band = zeros(windowed_bins_amount,1); % Make sure the function still works when the windowe bins amount is less than 3.
    else
        stop_band = [zeros(windowed_bins_amount-(w_length/2),1) ; w(1:(w_length/2))]; % create filter window by contagenating the window parts and zeros
    end
else
    ERROR('Choose a filter type, bs for band stop, lp for low pass and hp for high pass.');
end

frequency_data(startsample:endsample)=frequency_data(startsample:endsample).*stop_band; % filter using window
frequency_data(startsample_mirrored:endsample_mirrored)=frequency_data(startsample_mirrored:endsample_mirrored).*flip(stop_band); % filter mirrored spectrum using window

end
```

C.2.4. Windowing function

```
function [step_iteration , w] = window_function(w_length,window_choice)
% 03-06-2020
% window functions
% window choices are: 'hamming' 'hanning' 'rectangular'

%% The input
% The length of the window (w_length) and the type of window which is chosen (window_choice)

%% The output
% The length of the step (step_iteration) and the window (w)

%% copy code below to extract calculated window in main:
% window_choise = 'hamming'; % 'hanning' 'rectangular'
% [step_iteration , w] = window_function(N,window_choise); % call function
% w = [w;zeros(L-N,1)];

w = zeros(w_length,1); % fill the window vector with zeroes up to L for convenient implementation
switch window_choice
    case 'hamming' % Hamming Window "hammming"
        % Build the hamming window for fast implementation in analysis
        for n = (-ceil(w_length-1))/2:1:ceil(w_length-1)/2 % fill window according to defenition
            w(1+n+(ceil(w_length-1)/2)) = (25/46)+(1-25/46)*cos(2*pi*n/ceil(w_length-1));
        end
        step_iteration = ceil((w_length-1)/2); % to realise necessary overlap
```

```
    case 'hanning' % Hann Window "hanning"
        % Build the hann window for fast implementation in analysis
        for n = (-ceil(w_length-1))/2:1:ceil(w_length-1)/2 % fill window according
            to defenition
            w(1+n+(ceil(w_length-1)/2)) = 0.5+0.5*cos(2*pi*n/ceil(w_length-1));
        end
        step_iteration = ceil((w_length-1)/2); % to realise necessary overlap
    case 'rectangular' % Rectangular window "rectangular"
        % Buid the rectangular window
        w(1:w_length) = 1; % fill window according to defenition
        step_iteration = ceil(w_length); % no overlap allowed for rectangular func
    otherwise
        disp('Choose a build in window');
end
% window_param = [step_iteration ; w]; % fill return vector
end
```

C.2.5. F0 tracking

```
function [F0, Buffer, F0_cnt] = F0_tracking(Buffer, Fs, F0, difference_Hz, F0_cnt)
%% a function to track the fundamental frequency
% Johan Meyer, Folkert de Ronde.
% 04-06-2020
% This function tracks the fundamental frequency, it thus checks if it does
% not change too much in time.

%% Inputs
% the audio signal (Buffer), samplerate (Fs), fundamental
% frequency (F0), the maximum offset between jumps (difference_Hz) and the
% counted times that the jump was above the maximum (F0_cnt).

%% Outputs
% The fundamental frequency (F0), the frequency data (Buffer) and the
% counted times that the jump was above the maximum (F0_cnt).

    %% the function
    % determine the new F0
    F0_new = temp_F0_determination(Buffer,Fs);
    %difference_Hz = 5;

    %check if there is an original F0, if not then set the F0 to the new F0
    if F0 == 0
        F0 = F0_new;
    else
        %check if the original F0 and the new F0 lay appart too far, if so,
        %it must be a mistake
        if abs(F0 - F0_new) > difference_Hz
            F0_cnt = F0_cnt + 1;
            %if F0 and the new F0 lay apart too far for 4 times in a row,
            %then it is a jump instead of a mistake.
            if F0_cnt > 3
                F0 = F0_new;
                F0_cnt = 0;
            else
                F0 = F0; %illustration sentence, it bassically means that the value
                    does not change
```

```
            end
        else
            %F0 did not change too much, just a bit, must be correct, so
            %jump, also set the jump counter to 0.
            F0 = F0_new;
            F0_cnt = 0;
        end
    end
end
```

C.2.6. main Filter

```
%% Main filter
% Read audio file and filter just to see effect
% Folkert de Ronde, Johan Meyer
% 05-06-2020
% description:
% read files, process in latency dependent data blocks, zeropadd,
% f-resolution dependent zeropadd, window, amplitude filter, fft, find F0
% bandstopfilter, ifft, add and overlapp

close all;

%% Record an audio segment
[y, Fs, ~] = record_audio('filename', 3); % a function to record an audio segment

%% interpret loaded files and create result vectors
samples = length(y);     % determine number of samples
right = y(:,1); % right channel
right_edited = zeros(samples, 1); % empty vector to fill because otherwise this
    vector would never be the same length as 'samples'
left = y(:,2); % left channel
left_edited = zeros(samples, 1); % empty vector to fill because otherwise this
    vector would never be the same length as 'samples'

%% FFT parameters
latency = 10; % maximum acceptable latency due to buffering in milliseconds
f_delta = 10; % width of frequency bins, minimum defined by shift group
N = Fs*latency*10^-3; % determine samples per iteration
F0_fac = 4;  % number of repetitions in the buffer which is made for F0
z=0;
while F0_fac*N >= 2^z  % a while loop to determine the minimum value of L
    z=z+1;
end
L = 2^z;



%% Window choice time
window_choice = 'hanning'; % 'hanning' 'rectangular'
[step_iteration , w] = window_function(N,window_choice); % call function
w = [w;zeros(L-N,1)];

%% Window choice filter
window_choice = 'hamming'; % 'hamming' 'hanning' 'rectangular'
w_filter_length = 16;
```

```
[~, w_filter] = window_function(w_filter_length,window_choice); % call function

%% For loop to realisticly analyse data
start_sample = 1;                   % Entire spectrum
end_sample = length(y);             % Entire spectrum


buffer = zeros(L,1);                % Initialisation of the buffer variable

%% F0 tracking startup data
F0_cnt = 0;
F0 = 0;

for i = start_sample:step_iteration:end_sample-N
    buffer_iteration = ([right(i:i+N-1);zeros(L-N,1)]); % make the datapoints ready
        for fft by filling buffer

    %% Amplitude filter
    treshold = 0.003;       % The treshold for which an signal can be flagged as
        noise in V
    step_ampfilter= 100;    % The amount of samples which tested at the same time
    [buffer_iteration] = A_filter(treshold,step_ampfilter,N,buffer_iteration); %
        function call
    if sum(buffer_iteration) == 0 % Check if the data is all set to be noise
        noise = 1;
    else
        noise = 0;
    end

    if noise ~= 1  % If a set of samples is set to be noise, the filter does not
        process the samples
    %% Window
    buffer_iteration = buffer_iteration.*w; % Apply window to data of iteration

%% F0 buffer
    buffer(1:(F0_fac*N)-step_iteration) = buffer(1+step_iteration:(F0_fac*N));
    buffer(1+(F0_fac*N)-step_iteration:(F0_fac*N)) = zeros(1,step_iteration);
    buffer(1+((F0_fac-1)*N):((F0_fac-1)*N)+step_iteration) = buffer(1+((F0_fac-1)*N)
        :((F0_fac-1)*N)+step_iteration) + buffer_iteration(1:step_iteration);
    buffer(1+((F0_fac-1)*N)+step_iteration:((F0_fac)*N)) = buffer_iteration(1+
        step_iteration:N);
    buffer(1:step_iteration) = buffer(1:step_iteration) .* w(1:step_iteration);
%% FFT
    Buffer = fft(buffer);

%% Determine F0
    difference_Hz = 5;  % The amount of Hz which the F0 may step without the
        tracking filter doing anything
    [F0, Buffer, F0_cnt] = F0_tracking(Buffer, Fs, F0, difference_Hz, F0_cnt); %
        Function call

    %% filter around F0
    passband_div2 = 10; % The amount of Hz around the harmonics which is not
        filtered
    harmonic_skips = 0; % The amound of first harmonics which are ignored
```

```
89      [Buffer] = F0_dependent_filter(w_filter,Buffer, F0, L, Fs, passband_div2,
            harmonic_skips); % Function call
90
91      %% IFFT
92      inverse_buffer = real(ifft(Buffer)); % Fill inverse vector per iteration
93      inverse_buffer((F0_fac-1)*N+1:(F0_fac-1)*N+step_iteration) = inverse_buffer((
            F0_fac-1)*N+1:(F0_fac-1)*N+step_iteration) .* w(1:step_iteration);
94      right_edited(i:i+N-1) = right_edited(i:i+N-1)+inverse_buffer(((F0_fac-1))*N+1:(
            F0_fac)*N); % select desired data from the inverse_f0_buffer
95      end
96  end
```

C.2.7. Main filter, not sampled

```
1   %% Main filter
2   % Read audio file and filter just to see effect
3   % Folkert de Ronde, Johan Meyer
4   % 05-06-2020
5   % description:
6   % read files, process in latency dependent data blocks, zeropadd,
7   % f-resolution dependent zeropadd, window, amplitude filter, fft, find F0
8   % bandstopfilter, ifft, add and overlapp
9
10  close all;
11
12  [y,Fs] = record_audio(0,3);% Single frequency
13
14  %% interpret loaded files and create result vectors
15  samples = length(y);    % determine number of samples
16  right = y(:,1); % right channel
17  right_edited = zeros(samples, 1); % empty vector to fill because otherwise this
        vector would never be the same length as 'samples'
18  left = y(:,2); % left channel
19  inverse_left = zeros(samples, 1); % empty vector to fill because otherwise this
        vector would never be the same length as 'samples'
20
21  %% FFT parameters
22  latency = 10; % maximum acceptable latency due to buffering in milliseconds
23  f_delta = 10; % width of frequency bins, minimum defined by shift group
24  N = Fs*latency*10^-3; % determine samples per iteration
25  N = 512;
26  L = ceil(Fs/f_delta); % determine needed frequency bins te achieve freq resolution
27  L = 8192;
28
29
30  %% Window choice time
31  window_choise = 'hanning'; % 'hanning' 'rectangular'
32  [step_iteration , w] = window_function(N,window_choise); % call function
33  w = [w;zeros(L-N,1)]; % make the window the same length as the data
34
35  %% Window choice filter
36  window_choise = 'hamming'; % 'hamming' 'hanning' 'rectangular'
37  w_filter_length = 64;
38  [~, w_filter] = window_function(w_filter_length,window_choise); % call function
39
40  start_sample = 1;                   % entire spectrum
```

```
41 end_sample = length(y);              % entire spectrum
42 z=0;
43 while F0_fac*N >= 2^z  % a while loop to determine the minimum value of L
44     z=z+1;
45 end
46 L = 2^z;
47
48 buffer_f0 = zeros(L,1);
49
50 %% F0 tracking startup data
51 F0_cnt = 0;
52 F0 = 0;
53
54 buffer_iteration = ([right(1:1+N/2-1);right(1+N/2:1+N-1);zeros(L-N,1)]); % make the
       datapoints ready for fft by filling buffer
55
56 %% Amplitude filter
57  buffer_iteration = ([right(i:i+N-1);zeros(L-N,1)]); % make the datapoints ready for
        fft by filling buffer
58
59     %% Amplitude filter
60     treshold = 0.003;       % The treshold for which an signal can be flagged as
           noise in V
61     step_ampfilter= 100;    % The amount of samples which tested at the same time
62     [buffer_iteration] = A_filter(treshold,step_ampfilter,N,buffer_iteration); %
           function call
63     if sum(buffer_iteration) == 0 % Check if the data is all set to be noise
64         noise = 1;
65     else
66         noise = 0;
67     end
68
69     if noise ~= 1  % If a set of samples is set to be noise, the filter does not
           process the samples
70     %% Window
71     buffer_iteration = buffer_iteration.*w; % Apply window to data of iteration
72
73 %% F0 buffer
74     buffer(1:(F0_fac*N)-step_iteration) = buffer(1+step_iteration:(F0_fac*N));
75     buffer(1+(F0_fac*N)-step_iteration:(F0_fac*N)) = zeros(1,step_iteration);
76     buffer(1+((F0_fac-1)*N):((F0_fac-1)*N)+step_iteration) = buffer(1+((F0_fac-1)*N)
           :((F0_fac-1)*N)+step_iteration) + buffer_iteration(1:step_iteration);
77     buffer(1+((F0_fac-1)*N)+step_iteration:((F0_fac)*N)) = buffer_iteration(1+
           step_iteration:N);
78     buffer(1:step_iteration) = buffer(1:step_iteration) .* w(1:step_iteration);
79 %% FFT
80     Buffer = fft(buffer);
81
82 %% Determine F0
83     difference_Hz = 5;  % The amount of Hz which the F0 may step without the
           tracking filter doing anything
84     [F0, Buffer, F0_cnt] = F0_tracking(Buffer, Fs, F0, difference_Hz, F0_cnt); %
           Function call
85
86     %% filter around F0
```

```
87      passband_div2 = 10; % The amount of Hz around the harmonics which is not
            filtered
88      harmonic_skips = 0; % The amound of first harmonics which are ignored
89      [Buffer] = F0_dependent_filter(w_filter,Buffer, F0, L, Fs, passband_div2,
            harmonic_skips); % Function call
90
91      %% IFFT
92      inverse_buffer = real(ifft(Buffer)); % Fill inverse vector per iteration
93      inverse_buffer((F0_fac-1)*N+1:(F0_fac-1)*N+step_iteration) = inverse_buffer((
            F0_fac-1)*N+1:(F0_fac-1)*N+step_iteration) .* w(1:step_iteration);
94      right_edited(i:i+N-1) = right_edited(i:i+N-1)+inverse_buffer(((F0_fac-1))*N+1:(
            F0_fac)*N); % select desired data from the inverse_f0_buffer
95      end
```

C.2.8. peakfinder

```
1  function [loc] = peakfinder(Buffer_f0,seeking_length,F0)
2  %% This function finds the peaks of the F0
3  % 29-05-2020
4  % Johan Meyer & Folkert de Ronde.
5  % this code finds the peaks in a range around F0 to ensure the maximum
6  % frequency of F0.
7
8  %% The input
9  % The frequency data (Buffer_f0) the length in which the peak must be found (
       seeking_length) and the fundamental frequency (F0)
10
11 %% The output
12 % the location of the maximum
13
14 if F0+seeking_length > length(Buffer_f0)                                    %if the
       code tries to seek outside of the range of the length of Buffer_f0, then the code
        will seek no further then to the maximum length  of Buffer_f0
15     [~, loc] = max(Buffer_f0(F0-seeking_length:length(Buffer_f0)));
16 elseif F0-seeking_length < 1                                               % if the
       code stries to seek below the range of Buffer_f0, the code will seek from the
       first sample.
17     [~, loc] = max(Buffer_f0(1:F0+seeking_length));
18 else                                                                       %in this
       case the code seeks inside the limits of Buffer_f0, thus it works perfectly.
19     [~, loc] = max(Buffer_f0(F0-seeking_length:F0+seeking_length));
20 end
21 end
```

C.2.9. Temporary F0 determination

```
1  function [F0] = temp_F0_determination(Buffer_f0,fs)
2  %% F0 estimation algorithm
3  % in this algorithm F0 is determined by use of the complex cepstrum
4  % 25-05-2020
5  % made by Johan Meyer and Folkert de Ronde
6
7  %% The input
8  % the frequency data (Buffer_f0) and the samplerate (fs)
9
```

```
%% The output
% the fundamental frequency (F0)

%setting the starting values
dt = 1/fs;

%selfmade complex cepstrum
Buffer_log = log(abs(Buffer_f0));
cepstrum = real(ifft(Buffer_log));


%set time variable length
t_2 = 0:dt:length(Buffer_log)*dt-dt;

%set hz to seek between
trng = t_2(t_2>=10e-3 & t_2<=20e-3);
crng = cepstrum(t_2>=10e-3 & t_2<=20e-3);

%vind the peak between the set hz
[~,I] = max(crng);

%determine F0
F0 = 1/trng(I);
end
```

C.3. Testing

C.3.1. The audio recorder

```
% Audio recorder
% Folkert de Ronde en Johan Meyer
% 14-05-2020

% use function by calling: " record_audio('test',2) " in command line
% record_audio('Filename', SpeakingTime)
%  with SpeakingTime in seconds
% if "Filename" = 0 the file is not saved

function [y,Fs, recObj] = record_audio(filename,time)
%% record your voice
recObj = audiorecorder(16000,16,2); % (Fs, bits per sample, channels)
disp('Start speaking.')
record(recObj)
pause(time)
stop(recObj)
disp('End of Recording.');

y = getaudiodata(recObj); % load the left and right channel from recObj
Fs = recObj.SampleRate;  % load sample rate
BitsPerSample = recObj.BitsPerSample; % load bits per sample

if filename ~= 0
    save(filename, 'y', 'Fs', 'BitsPerSample'); % save workspace
end
end
```

C.3.2. Test program

```
%% Main filter
% Read audio file and filter just to see effect
% Folkert de Ronde, Johan Meyer
% 05-06-2020
% description:
% read files, process in latency dependent data blocks, zeropadd,
% f-resolution dependent zeropadd, window, amplitude filter, fft, find F0
% bandstopfilter, ifft, add and overlapp

close all;

%% Declare names of .WAV audiofiles for easy loading and testing
pre = 'premeting_21814427_29-11-18.wav';
post3mnd = 'post_3_mndn.wav';
post6mnd = 'post_6_maanden.wav';
post1jr = 'post_1_jaar.wav';

%% Declare names of .MAT testaudio files
test = 'test.mat';
zin = 'zin.mat';
a = 'a_aanhouden.mat';
af0 = 'aaF0.mat';
doremi = 'doremi.mat';
abc = 'abc_tvnoise';
alh = 'a_laaghoog.mat';
test_files = ["a", "o", "u", "e"];
%% Read files, comment inapplicable line
% .WAV files
%[y,Fs] = audioread(post3mnd); % load audiofile and sample frequency
% .MAT files
load(a); % load wrokspace from audio file containing y, Fs, BitsPerSample
%load(alh); % load wrokspace from audio file containing y, Fs, BitsPerSample

%[y,Fs] = record_single_frequency(40,3);% Single frequency
%[y, Fs, ~] = record_audio('zin_folkert', 3);  % Record an audio sample
%real time.
%for o = 1 : length(test_files)   % a for loop which is used to display
%different syllables
%load('a')

%% interpret loaded files and create result vectors

clean = y(:,1); % store the clean result for testing
samples = length(y);
SNR = 19; % Determine the SNR.
y = awgn(y,SNR,'measured'); % edit the clean speech, to have noise in it
right = y(:,1); % right channel
inverse_right = zeros(samples, 1); % empty vector to fill because otherwise this
    vector would never be the same length as 'samples'
left = y(:,2); % left channel
inverse_left = zeros(samples, 1); % empty vector to fill because otherwise this
    vector would never be the same length as 'samples'

F0_vect = zeros(samples,1); % Add a test vector to make a plot with all the F0's
I_o = zeros(50-7,1);        % initialize I
```

```
54
55 %% FFT parameters
56 % for latency = 4:50
57 latency = 10; % maximum acceptable latency due to buffering in milliseconds
58 f_delta = 10; % width of frequency bins, minimum defined by shift group
59 N = round(Fs*latency*10^-3); % determine samples per iteration
60 F0_fac = 4;
61 z=0;
62 while F0_fac*N >= 2^z
63     z=z+1;
64 end
65 L = 2^z; % Set L to it's minimal needed value
66
67
68 %% Window choice time
69 window_choice = 'hanning'; % 'hanning' 'rectangular'
70 [step_iteration , w] = window_function(N,window_choice); % call function
71 w = [w;zeros(L-N,1)];
72
73
74
75 %% For loop to realisticly analyse data
76 %start_sample = 45007;              % guus speech
77 %end_sample = 45007+20*N;           % guus speech
78 % start_sample = 41440;             % folkert speech
79 % end_sample = start_sample+N*5;    % folkert speech
80 start_sample = 1;                  % entire spectrum
81 end_sample = length(y);            % entire spectrum
82 buffer = zeros(L,1);
83
84 %% F0 tracking startup data
85 F0_cnt = 0;
86 F0 = 0;
87
88 %% set startup data for I
89 max_freq = 22; % Maximum passband frequency
90 I = zeros(max_freq,1); % initialize values
91 maximum_I_per_q = zeros(length(I)); % initialize values
92 %% Window choice filter
93 window_choise = 'hamming'; % 'hamming' 'hanning' 'rectangular'
94 w_filter_length_choose = [2,4,8,16,32,64,128]; % add a vector of different filter
       lengths
95 for q = 1:length(w_filter_length_choose)
96 w_filter_length = w_filter_length_choose(q); % Loop through the filter lengths
97 [~, w_filter] = window_function(w_filter_length ,window_choise); % call function
98
99
100 for passband_div2  = 1:max_freq  %loop through the frequencys
101
102 for i = start_sample:step_iteration:end_sample-N % modifie the data as always
103     buffer_iteration = ([right(i:i+step_iteration-1);right(i+step_iteration:i+N-1);
          zeros(L-N,1)]); % make the datapoints ready for fft by filling buffer
104
105     %% Amplitude filter
106 %     treshold = 0.003;
107 %     step_ampfilter= 100; %ceil(L/10);
```

```
%     [buffer_iteration] = A_filter(treshold,step_ampfilter,N,buffer_iteration);
%     if sum(buffer_iteration) == 0 %check if the data is all set to be noise
%         noise = 1;
%     else
%         noise = 0;
%     end
%
    %if noise ~= 1
    %% Window
    buffer_iteration = buffer_iteration.*w; % apply window to data of iteration

%% F0 buffer
    buffer(1:(F0_fac*N)-step_iteration) = buffer(1+step_iteration:(F0_fac*N));
    buffer(1+(F0_fac*N)-step_iteration:(F0_fac*N)) = zeros(1,step_iteration);
    buffer(1+((F0_fac-1)*N):((F0_fac-1)*N)+step_iteration) = buffer(1+((F0_fac-1)*N)
        :((F0_fac-1)*N)+step_iteration) + buffer_iteration(1:step_iteration);
    buffer(1+((F0_fac-1)*N)+step_iteration:((F0_fac)*N)) = buffer_iteration(1+
        step_iteration:N);
    buffer(1:step_iteration) = buffer(1:step_iteration) .* w(1:step_iteration);

    Buffer = fft(buffer);
%% Determine F0
    difference_Hz = 5;
    [F0, Buffer, F0_cnt] = F0_tracking(Buffer, Fs, F0, difference_Hz, F0_cnt);

    %% filter around F0
%     passband_div2 = 15;
    [Buffer] = F0_dependent_filter(w_filter,Buffer, F0, L, Fs, passband_div2,0);
    %% shift function
    % use shifted_Y to store shifted window
    % the result is later stored in the inverse_shifted vector
    %% IFFT
    inverse_buffer_f0 = real(ifft(Buffer)); % fill inverse vector per iteration
    inverse_buffer_f0((F0_fac-1)*N+1:(F0_fac-1)*N+step_iteration) =
        inverse_buffer_f0((F0_fac-1)*N+1:(F0_fac-1)*N+step_iteration) .* w(1:
        step_iteration);
    inverse_right(i:i+N-1) = inverse_right(i:i+N-1)+inverse_buffer_f0(((F0_fac-1))*N
        +1:(F0_fac)*N); % select desired data from the inverse_f0_buffer
    %end
end
    I(passband_div2) = SIIB(clean(start_sample:end_sample),inverse_right(
        start_sample:end_sample),Fs); % Store the I for every frequency
    %I_o(latency-3) = SIIB(clean(start_sample:end_sample),inverse_right(start_sample
        :end_sample),Fs); %Store I for every latency
end
maximum_I_per_q(q) = max(I); % Store the maximum value of I per window length
%d = stoi(right(start_sample:end_sample),inverse_right(start_sample:end_sample),Fs);
hold on
title('I vs the passband range for different window lengths')
plot((1:passband_div2),I)
ylabel('I [b/s]')
xlabel('passband range [Hz]')


% plot((4:latency),I_a)
% hold on
```

```
% plot((4:latency),I_o)
% plot((4:latency),I_u)
% plot((4:latency),I_e)
% I_avg = (I_a+I_o+I_u+I_e)/4;
% plot((4:latency),I_avg)
% ylabel('I [b/s]')
% xlabel('latency [ms]')
% title('I vs the latency')
end
Legend=cell(length(w_filter_length_choose),1);
 for iter=1:length(w_filter_length_choose)
   Legend{iter}=['windowlength = ', num2str(w_filter_length_choose(iter))];
 end
 legend(Legend)
% legend('letter = a', 'letter = o', 'letter = u', 'letter = e', 'The average of all
    letters');
maximum_I = max(maximum_I_per_q);

%% Plot frequency domain of original and filtered signal in 1 plot
%freq_plotter(end_sample-start_sample+1,Fs,y(start_sample:end_sample,1)',
    inverse_right(start_sample:end_sample)',inverse_right(start_sample:end_sample)');
     % plot frequency spectrum

%% Store files in audio player to hear result
x = [y(start_sample:end_sample,1);inverse_right(start_sample:end_sample)];
origin = audioplayer(y(start_sample:end_sample,1),Fs); % store original sound in
    player to listen
aud = audioplayer(inverse_right(start_sample:end_sample),Fs); % store altered sound
    in player to hear result
test = audioplayer(x,Fs);
shifted = audioplayer(inverse_shifted(start_sample:end_sample),Fs);
```

C.4. ADC

C.4.1. SNR determination

```
%% an algorithm to determine the SNR of an ADC
% Johan Meyer, Folkert de Ronde
% 19-06-2020

%% The function
number_of_bits = 8;
operating_voltage = 3;
max_error = operating_voltage/(2^number_of_bits); % in volts
SNR = 20*log10(operating_voltage/max_error); % in dB
```

D | Statement of requirements

Below a translation of the statement of requirements agreed upon with the client.

D.1. Assignment description

The voice amplification device is part of a second generation aid device for Laryngectomised People (LP) called the Exobreather. This device conditions air (filter, heat and moisturises) what would normally have been done by the nose mouth and throat cavities. Furthermore a hands free speech utility and a stoma fluid absorption utility are in the pipeline to reduce stoma mainenance to once a day. This device is worn beneath the clothing, below the stoma on the upper chest. The voice amplification device is, including all its peripheral equipment, part of the Exobreather. For this reasin the complete design should be as compact as possible.
The system should be an all in one package with preferentially no loose parts. For that reason a microphone should be attachable to the Exobreather (EB). The speakers should preferentially be integrated into the Exobreather and for that reason we wearable below the clothing. The on off switch and volume control could be controlled by a pressure sensor in the stoma.
The current available hands free systems work by giving an air pressure shock, closing the stoma. This system works quite intuitively in practise. In the client case the speech pressure range starts from 150mm water column and at 700 mm water column the clients speaks at full capacity.
The system does not need to be on at all times and should jump in when the user raises their voice. A suitable volume control is to be determined.

D.2. Statement of requirements

D.2.1. General requirements

- The System comprises of and suitable components are selected for:
  - A microphone
  - A signal processing part
  - An amplifier
  - A speaker
  - An energy source
- It is wearable on the body
- Due to the current corona health crisis a functioning prototype is not part of the SOR
- Its operating temperature stays below $36^{\circ}C$
- The input bandwidth is at least 50 Hz to 4 kHz
- The output sound intensity is at least 85 dB at 0.3 m
- The output bandwidth is at least 50 Hz to 4 kHz
- The microphone can not be a headset
- The volume control is hands free
- The system should provide 2 hours of speaking time,
- of which 30 minutes at high sound intensity
- The device should be rechargeable,
- with maximally 8 hours of charging time

D.2.2. Signal processing requirements

- Background noise is suppressed by 6 dB
- Acoustic feedback should be avoided
- The F_0 should be shifted up to resemble a less pathological voice
- The voice produced should be intelligible

D.2.3. Additional preferences

- The total costs of the parts should be around 100 euros
- It is integratable with the 'Exobreather'
- The size will have to be as limited as possible
- The latency of sound produced should stay below 15 ms
- The reproduced voice should be as natural as possible
- The volume should be regulated by air pressure

D.2.4. Closing statement

At the end of the project trajectory an extensive specification of the voice enhancement device / voice correction device is delivered. Another research group should be able to continue the research using the results of this preliminary investigation.
An NDA is part of the assignment.

D.3. Original statement of requirements

For completeness the original SOR is included below in Dutch.

Opdrachtformulering voor BAP studenten groep I

05-05-2020, Delft

Opdracht omschrijving:

Stem versterking voor mensen zonder stembanden:
De stemversterker maakt deel uit van een 2de generatie hulpmiddel voor LP Exobreather genaamd
Dit apparaat verzorgt het conditioneren van de lucht (reinigen, opwarmen en bevochtigen) dat anders in je neus mond keelholte gebeurt. Tevens komt er een voorziening zodat je handsfree kan spreken en er komt een slijmvanger zodat je maar een keer per dag je stoma etc hoeft te verzorgen. Dit apparaat wordt onder de kleding onder de stoma op het bovendeel van de borst gedragen. Het spraakversterker systeem maakt inclusief alle accessoires onderdeel uit van de Exobreather. Daarom moet alles zo compact mogelijk worden gebouwd.
We gaan voor een all-in one systeem dus bij voorkeur geen losse delen.
We gaan daarom voor een microfoon die dus aan de Exobreather (EB) bevestigd kan worden.
De speakers moeten bij voorkeur ook in de Exobreather worden ondergebracht die verdwijnen bij voorkeur dus ook onder de kleding. Het aan en uit schakeling en de volume control zou dmv een druksensor die in de stoma zit kunnen geschieden.
De handsfree systemen die je nu kan gebruiken werken namelijk als volgt: je geeft even met je adem een drukstoot dan sluit een klep de stoma af. En in de praktijk werkt dit behoorlijk intuitief.
In mijn geval kan ik beginnen met praten bij circa 150 mmWK en bij 700 mmWK presteer ik maximaal.
Het systeem hoeft niet altijd aan te staan het moet bijspringen zo gauw als je je stem wil verheffen er wordt nog een gepaste volumeregeling bepaald.

Programma van eisen

- Algemeen
    - Het systeem bestaat uit een microfoon, signaal bewerk systeem, versterker, speaker, energiebron, te dragen op het lichaam
    - Het leveren van een werkend prototype hoort vanwege de corona niet meer tot de opdracht.
    - De bedrijfstemperatuur blijft beperkt tot 36 °C
    - Geluid input spectrum minimaal 50Hz tot 4000Hz
    - Maximaal uitgangsgeluid minimaal 85dB at 0.3 meter
    - Frequentie bereik uitgang systeem 50Hz tot 4000Hz
- Microfoon
    - Selecteren van de microfoon
    - Geen headset
- Signaal bewerk systeem
    - Filter achtergrond geluid 6 dB
    - Filter rondzingen
    - Verschuif de toonhoogte naar die van een gezonde stem (statisch / dynamisch)
    - Stemgeluid moet verstaanbaar zijn
- Speakers
    - Selecteren van de speaker
- Versterker

    - Selecteren van de versterker
- Aan/uit en volume regel systeem
    - Volume regeling is hands free
- Energie bron
    - Opstellen van specificaties voor energiebron (inschatten van opslagcapaciteit)
        - Levensduur systeem = 2 uur op een dag spreken waarvan 30 min hard
        - Oplaad tijd = 8uur
    - Selectie van het meest geschikte type energiebron

Additionele wensen

- De totaalkosten richtlijn zal 100 euro bedragen
- Systeem wordt integreerbaar met de exobreather
- Het formaat blijft zo beperkt mogelijk
- Latency 15 ms
- Stemgeluid moet natuurlijk blijven
- Volume regelbaar op luchtdruk

AAN HET EIND VAN JULLIE ONDERZOEK LEVER JE EEN TO THE POINT UITGEBREIDE SPECIFICATIE VAN DE STEMVERSTERKER/STEMCORRECTIESYSTEEM WAAR DE “AFDELING” NA JULLIE WEER MEE VERDER KAN MET DE RESULTATEN VAN JULLIE VOORONDERZOEKINGEN

Een NDA is onderdeel van de opdracht.

,