LOW-COST HEARING AID

TABLE OF CONTENTS

1.ABSTRACT

Hearing aids are primarily useful in improving the hearing and speech comprehension of people who have hearing loss that results from damage to the small sensory cells in the inner ear, called hair cells. Approximately 10% of the world's population suffers from some type of hearing loss, yet only a small percentage of this statistic use a hearing aid. The main reason for the low usage is due to the high cost. So, in order to overcome the barrier of high cost, we have come up with the low-cost hearing aid. Through the use of digital signal processing, digital hearing aid now offers what the analog hearing aid cannot offer. In this project, the prototype of simple digital hearing aid is developed using Python programming language with the help of Raspberry Pi as MCU. The implementation of this configurable digital hearing aid system includes the noise reduction filter, frequency shaper function, and amplitude compression function. This digital hearing aid system is design to adapt for mild and moderate hearing loss patient since different gain can be set to map different levels of hearing loss.

We are using Raspberry Pi 3 as a processor for computation and processing of the signal. Here real time speech is given as an input for further processing. The input is given through USB mic and further processing is done using Raspberry Pi 3. The further processing includes noise reduction, amplification of the input voice signal, frequency shaping and amplitude compression. The processed signal is our output signal which is transmitted using Bluetooth. The person with hearing imparity will wear the Bluetooth device which act as hearing aid. This hearing aid aids in improved ease in listening environments, Less interference from a moderate amount of background noise, Affordable Improved hearing quality at low cost. This product can be used by the person with partial hearing losses. The cost of analog hearing aid is too high, which a common man cannot afford. In order to make it available to the persons with partial hearing problems we are aiming to develop digital hearing aid using signal processing and Raspberry pi 3b+.

2.Introduction

A hearing aid is a small electronic device that people wear in or behind their ears. It makes some of the sounds louder so that a person with hearing loss can listen, communicate, respond and participate more fully in daily activities. It also helps the other person not to waste his energy shouting in order to make his words listen to the person with hearing impairment. A hearing aid can help people hear more in quiet and noisy situations. Hearing aids are primarily useful in improving the hearing and speech comprehension of people who have hearing loss that results from damage to the small sensory cells in the inner ear, called hair cells. Approximately 10% of the world's population suffers from some type of hearing loss, yet only a small percentage of this statistic uses a hearing aid. The main reason for the low usage is the high cost.

Now whenever a person with a hearing impairment wants to buy a hearing aid then the shopkeeper gives him two options, whether he wants analog hearing aid or a digital one? Both of them perform the amplification of the sound and give a distortion-free sound with clarity, then why analog and digital?

The analog hearing aid has three basic sections. 1) Microphone: which takes input from the user, 2) pre-amplifier or amplifier- to amplify the sound and 3) Receiver or speaker- which delivers the amplified signal to the human ear. It performs the function as required but the disadvantage is that feedback cannot be properly controlled and background noise level cannot be properly controlled in analog hearing aids. Digital hearing aid takes the input through the microphone which is analog data and converts that to digital then processes it, and produces the output in analog form. The main advantage associated with this is feedback is properly controlled. It identifies which is feedback and which is high pitch sound. The processor cancels the feedback so that the listener can clearly hear.

Using analog hearing it is not possible to separate wanted and unwanted noise. For example, it is best suited for a person who speaks in a quiet room but if there are a lot of background noises then it considers all of it and it is also not possible to separate the noise. These problems are overcome using the digital hearing aid. Here it is possible to reduce the unwanted noise, amplify the sound and filter the real-time data.

We also know that the cost of digital hearing is very high that a normal middle house people cannot afford it. Thus, most people go for analog. In order to make the common people afford the digital hearing aid, we have developed a low-cost digital hearing aid.

3.Objectives:

- To develop a digital hearing aid that performs almost the same function as the one which is available on the market.
- To make every person who partially lost his hearing to afford the Hearing aid.
- To make use of the same Hearing aid kit for his entire lifetime.

4.Methodology:

The first phase of the project started by searching for the algorithms which collect the real-time data. We choose Python to develop our project because several packages are available in python, particularly for signal processing. In this project the packages which we used are

4.1 Libraries Used:

4.1.1 Sounddevice:

As stated in its documentation, python-sounddevice “provides bindings for the PortAudio library and a few convenience functions to play and record NumPy arrays containing audio signals”. In order to play WAV files, numpy and soundfile need to be installed, to open WAV files as NumPy arrays. In our project this library helps in getting the input and transferring the output.

4.1.2 NumPy:

NumPy is the most important Python package for scientific computing. It's a Python library that includes a multidimensional array object, derived objects, and a variety of routines for performing fast array operations, such as mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and more. In this project, the real-time data is processed as a one-dimensional array by the raspberry pi, we imported this package.

4.1.3 Argparse:

Writing user-friendly command-line interfaces is simple by using the argparse package. The program specifies the arguments it requires, and argparse deduces how to extract them from sys. argv. When users provide the application with invalid parameters, the argparse module automatically generates help and usage messages and raises errors. In this project, argparse helps in accessing the ports in the system which is the functionality provided by sys.argv.

4.1.4 SciPy:

SciPy is a library in Python for scientific and technical computing. Optimization, integration, interpolation, eigenvalue issues, algebraic equations, differential equations, statistics, and many other types of problems are all covered by SciPy. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The SciPy library was designed to operate with NumPy arrays and includes a number of user-friendly and efficient numerical procedures, such as numerical integration and optimization methods. They work on all major operating systems, are simple to install, and are completely free. SciPy is mainly used for filtering purposes. Noise is the main barrier when we are processing real-time systems. This library can be used to remove the noise which gets added to the input.

4.2 Hardware Specifications:

4.2.1Raspberry Pi Model3B+

Fig 1. Raspberry Pi Model 3B+

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT.

The secret sauce that makes this computer so small and powerful is the Broadcom BCM2837, an ARM Cortex-A53 64-bit Quad-Core Processor System-on-Chip operating at 1.4GHz. The GPU provides OpenGL ES 2.0, hardware-accelerated OpenVG and 1080p30 H.264 high-profile decode. It is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of a general-purpose computer. What does that all mean? It means that if you plug the Raspberry Pi 3 B+ into your HDTV, you could watch Blu-ray quality video, using H.264 at 40MBits/s.

Features of RPI 3B+:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

4.2.2 Raspberry Pi USB Mic:

Fig2. Raspberry Pi USB Mic

Raspberry Pi USB Plug and Play Desktop Microphone is USB Microphone that offers compatibility with any plug-and-play enabled Raspberry Pi Model B+, 2 model B, Raspberry Pi 3 as wells as it is also Compatible with PC and Mac, Ideal for Chatting on Skype for video chat or useful for the recording of sound. The Mic has advanced digital USB provides superior clarity with the simplicity of a single USB plug – and – play connection. Microphone pivots on base to hold preferred position.

Features of Raspberry Pi USB Mic:

- Sensitivity: -47dB ± 4dB
- Sensitivity reduction: -3dB at 1.5V
- Working Voltage: 4.5V
- Frequency response: 100 ~ 16kHz
- SNR: More than -67dB
- Cable length: 0.7-0.9M
- The effective distance of 2 meters or more

Then for the transmission of the amplified and filtered audio signal which will be heard by the hearing-impaired person we use the Bluetooth Headset which are compatible to the mobile phones. Bluetooth protocol which is inbuilt in the Raspberry Pi supports the transmission of the processed data.

4.3 Flowchart:

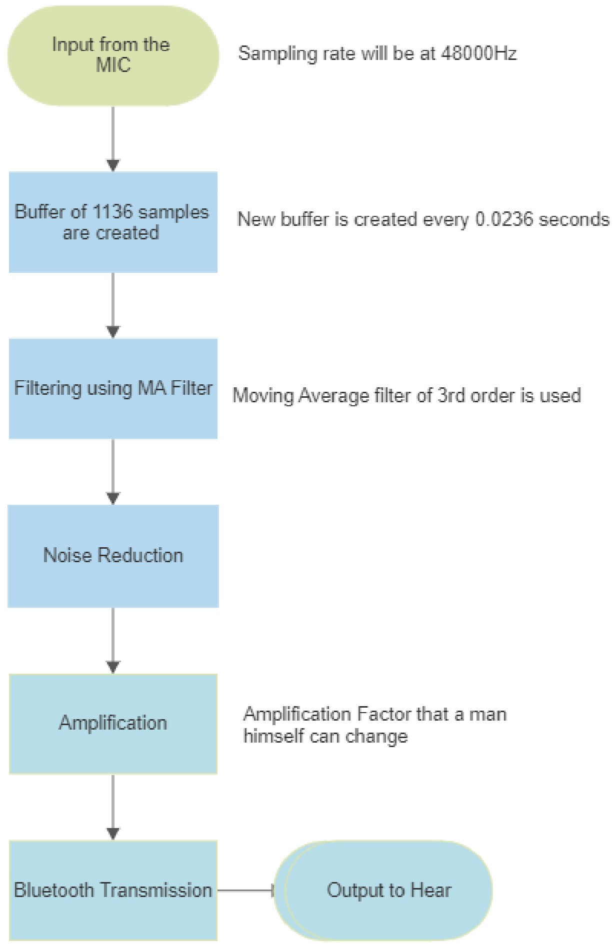

Fig3. Flowchart of the system

4.4 Block Diagram:

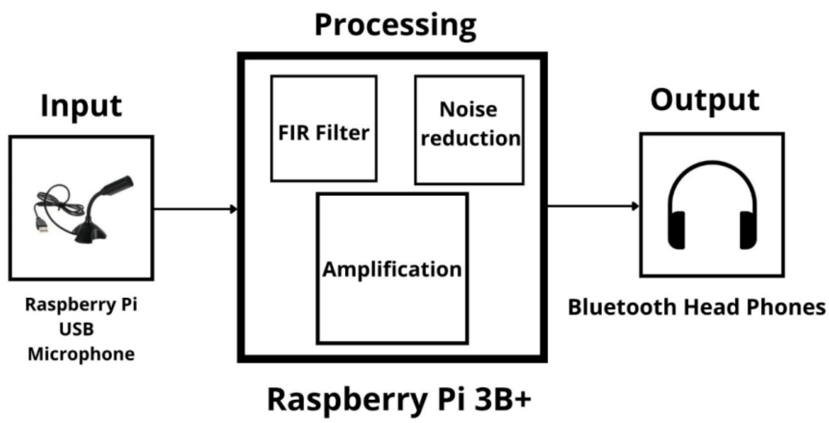

Fig4. Block Diagram of the System

4.5 Process:

The input will be fed to the mic which is the normal speech signal from a person with whom the hearing-impaired person will be communicating. The signal will be a raw input for the Processing unit. Here for the processing of the signal we are using Raspberry Pi Model 3B+. The sampling rate of the USB mic what we are using is of 48000 Hz. The input speech signal is temporarily stored into a buffer i.e., an array over here which will be easy for processing in Python.

To go deeper in the concept of buffer, here the buffer size is of 1136 with 2-Dimensional Array because the input is Dual channel Audio. In further processing the data will be converted to 1-Dimensional Array which will be easy in computing and convoluting. New buffers are created continuously for every 0.0236 seconds.

The created buffer is taken as a variable for processing. The first process that is deployed is Moving Average Filter. The Moving average filter abruptly tries to reduce the surrounding noise and helps in smoothening the speech signal. Smoothening of the speech signal is an advantage over here because if a person shouts or a noise of high amplitude arises i.e., we may consider it as a bust noise, then smoothening for some extent can reduce the damage that happens to a person’s ear. The filtering and noise reduction part is completed over here.

Later, we have the Amplitude modification. Here in amplitude modification, this part can be adjusted as per persons need. So that the same Hearing aid kit can be used for his entire lifetime that means by adjusting the amplitude on ageing.

Later the output is transmitted through Bluetooth to the headset which the person is wearing.

5. Results and Discussions:

5.1 Using the appropriate Filter:

After comparison with the FIR filter and IIR filter, it was chosen to use the Moving Average (MA) Filter. The moving average filter is a special case of the regular FIR filter. Both filters have finite impulse responses. The moving average filter uses a sequence of scaled 1s as coefficients, while the FIR filter coefficients are designed based on the filter specifications. They are not usually a sequence of 1s.

The moving average of streaming data is computed with a finite sliding window:

$$movAvg = \frac{x[n] + x[n-1] + ... + x[n-N]}{N+1}$$

N + 1 is the length of the filter. This algorithm is a special case of the regular FIR filter with the coefficients vector, [b0, b1, ..., bN]

$$FIROutput = b_0x[n] + b_1x[n-1] + ... + b_Nx[n-N]$$

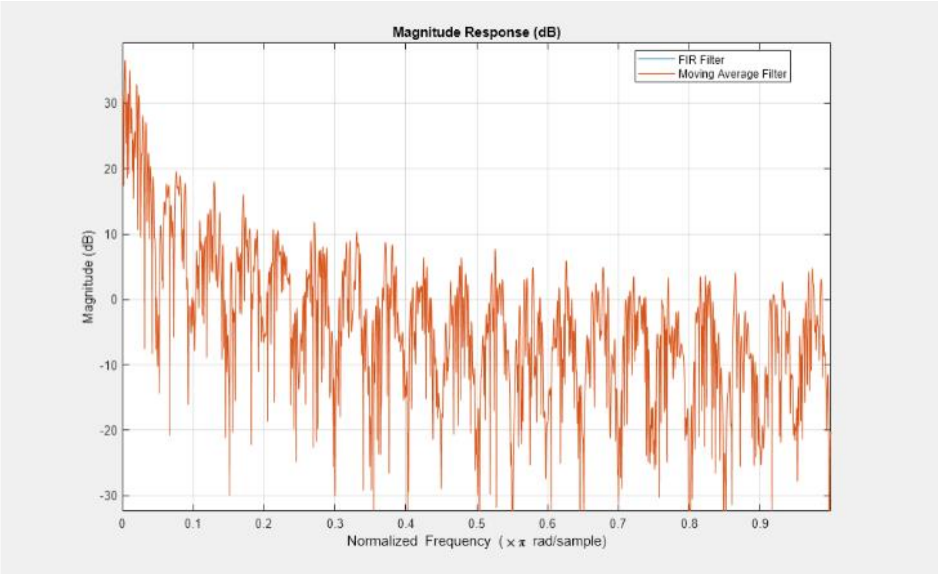

Fig5. Frequency response of the FIR Filter

Compare the filter's frequency response to that of the ideal filter. You can see that the main lobe in the passband is not flat and the ripples in the stopband are not constrained. The moving average filter's frequency response does not match the frequency response of the ideal filter.

To realize an ideal FIR filter, change the filter coefficients to a vector that is not a sequence of scaled 1s. The frequency response of the filter changes and tends to move closer to the ideal filter response. Design the filter coefficients based on predefined filter specifications. For example, design an equiripple FIR filter with a normalized cut-off frequency of 0.1, a passband ripple of 0.5, and a stopband attenuation of 40 dB.

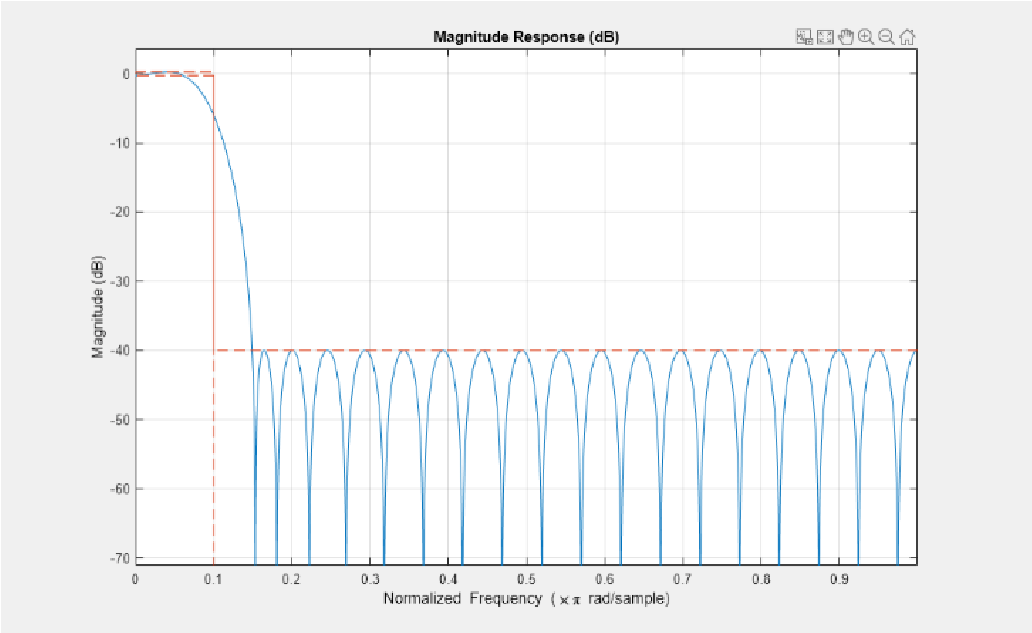

Fig6. Magnitude Response of MA filter

The filter's response in the passband is almost flat (similar to the ideal response) and the stopband has constrained equiripples.

By making all these comparisons the Moving average filter was used. FIR filter designing for a system having real time noise was a tedious job so MA Filter which matches the frequency and magnitude response of itself to the Frequency and Magnitude response of the noise present in the Real time speech signals.

At first the MA Filter was tested on a WAV file containing Noise and comparison was made between filtered signal with the noise added signal.

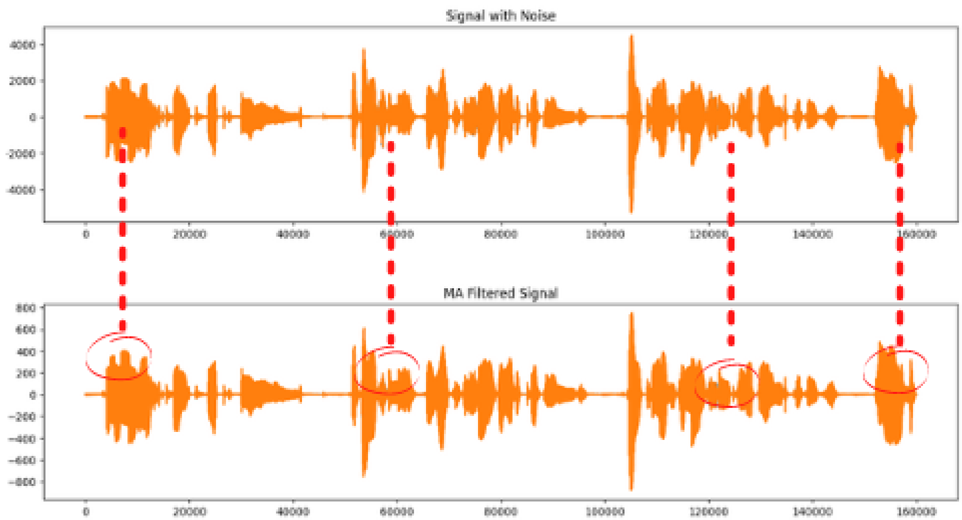

Fig7. MA Filter testing on a Sample Audio file

Amplification can be done on the requirements of the person by programming according to the persons need.

```
outdata[:] = indata*2
```

Code Snippet

Here 2 is the amplification factor that has been considered. This can be varied according to a persons need.

5.2 Cost Calculation

Components	**Cost**
Raspberry Pi Model 3B+	4500 INR
Raspberry Pi USB Mic	250 INR
Bluetooth Headset	1500 INR
MicroSD card for Raspbian	250 INR
Total	**6500 INR**

Table1. Cost estimation Table

So by comparing All these means we can say this project would be socially useful for the people.

6. CONCLUSION & FUTURE WORK

6.1 Conclusion:

We proposed a Low-Cost Hearing Aid, which yields the best result as same as that of the digital hearing aids that are available in the market. The filter used in this hearing aid is capable of filtering the signal close to the characteristics of the ideal filter. The variable amplitude is the plus point that adds to this aid because the person need not change the hearing aid upon ageing. Even we can achieve the development of complete setup at low cost so that all middle class people who had the barrier of hearing can afford.

6.2 Future Work:

The future work includes complete fabrication of the system just by using a Microprocessor which will be miniature in size and mounting of wireless mic to the system makes the system around 95% compact. We also aim to develop the in-Lube Hearing system by fabrication of the Microprocessor Chip. The hearing aid will be embedded on the back of the ear so that the highly sensitive mic will be placed at human height. Adding further we can use a better filter with a good frequency response.

7. Reference:

- https://earguru.in/hearing-aids/should-i-buy-an-analogue-or-a-digital-hearing-aid/
- https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_ch15.pdf
- https://www.researchgate.net/publication/330703731_Comparative_Study_of_FIR_Digital_Filter_for_Noise_Elimination_in_EMG_Signal
- https://www.ee.iitb.ac.in/~esgroup/es_mtech02_sem/es02_sem_rep_arojit.pdf
- https://www.sciencedirect.com/topics/engineering/moving-average-filter
- http://oops.uni-oldenburg.de/331/1/361.pdf
- https://codemonk.in/blog/moving-average-filter/