Powerline Interference using Notch and Comb Filter

Kajith.T
CB.EN.U4CCE22050
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India

Sukkesh
CB.EN.U4CCE22051
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India

Lithika.S
CB.EN.U4CCE22053
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India

***Abstract*—This research addresses the task of improving the quality of Electrocardiogram (ECG) signals by mitigating the impact of powerline interference (PLI) noise. Traditional notch filters often prove ineffective due to the dynamic frequency nature of PLI. The proposed method involves the use of a cascade wide-band notch filter, showcasing improved accuracy in ECG signal processing and a reduction in wide-band electrical noise. The transfer function (H(z)) serves as a mathematical representation of the relationship between input and output signals in discrete-time systems. this transfer function signifies a system with zero, where the zeros define frequencies at which the system response is null. This parameter-free algorithm not only efficiently reduces PLI but also preserves QRS-complex features in the filtered signal, offering a promising solution for addressing challenges in ECG signal de-noising. Furthermore, the study explores the integration of comb filters, strategically designed for targeting powerline frequencies and odd harmonics, as an additional approach for interference reduction, specifically tailored to common frequencies such as 50 or 60 Hz,180 Hz,300Hz,..etc**

Keywords— Notch filter, transfer function(H(z)), PLI, Comb filter, Fast Fourier Transform, Signal De-noising

INTRODUCTION

The Electrocardiogram (ECG) stands as a widely employed non-invasive method for measuring heart rhythms through electrical activity on the skin's surface. Despite its ubiquity, the ECG signal poses challenges due to its low strength and vulnerability to various noise sources, especially during recording. The translation of heart rhythms into electrical activity is prone to corruption by diverse noise types, including baseline offset, electrode contact noise, muscle state, motion artifacts, instrumentation, and power line interface noise (PLI). This discussion hones in on the specific challenge of PLI and its associated harmonics intertwined with the ECG signal, addressed through the utilization of special-purpose filters.

In existing literature, numerous FIR and IIR filters have been proposed to mitigate PLI. However, the real-time recording of the ECG signal introduces unique challenges as the frequency fluctuates narrowly around the 60 Hz fundamental frequency. This variation stems from differences in available power supplies adhering to distinct standards, resulting in a frequency range between 50-60 Hz. Consequently, there arises an intuitive need for a special-purpose filter designed to eliminate PLI frequencies within the 47-53 Hz band while preserving the vital frequencies of the ECG signal in that range to prevent degradation. In such scenarios, a special-purpose filter becomes imperative, with its response automatically adjusted through a few variable parameters per specified criteria. Typically, an IIR notch filter, with a 3dB rejection band, is applied to isolate sinusoidal and broadband components.

Notch filters find varied applications, and one of their crucial uses is in reducing power line interference, considering the power line as a significant source of noise. Cables transmitting ECG signals prove highly sensitive to electromagnetic interference (EMI) originating from patients and monitoring equipment. Over time, several strategies have been suggested to eliminate power line interference during ECG, making the removal of such interference through a notch filter a vital area of research. The misinterpretation of the ECG signal due to power line interference could lead to significant issues in the medical field. This paper specifically delves into the design of a notch filter tailored to eliminate undesirable PLI from the ECG signal at 60, 180, and 300 Hz noise harmonics.

In the array of tools available to mitigate interference, comb filters emerge as a noteworthy solution. A comb filter operates as a frequency-selective filter designed to suppress signals at specific frequencies while permitting others to pass through. In the context of PLI, these filters prove particularly effective in addressing interference stemming from powerline frequencies and their harmonics. What sets comb filters apart is their systematic attenuation of unwanted frequencies achieved by strategically placing notches at the powerline frequency and its harmonics. This introductory concept lays the foundation for exploring the design, implementation, and application of comb filters as practical and efficient tools in combating powerline interference. With electronic systems continuing to proliferate in powerline-rich environments, the significance of comb filters in enhancing signal integrity becomes increasingly apparent.

Expressed as a rational function of the complex variable \(z\), the transfer function encapsulates the system's dynamics, offering engineers and researchers a tool for analyzing, designing, and optimizing digital systems. Zeros and poles in the transfer function align with frequencies where the system's response is null or unbounded, respectively. Stability, frequency response, and transient characteristics represent essential facets explored through the transfer function in the Z-domain. As digital signal processing technologies advance, the transfer function in the Z-domain stands as a cornerstone for the development and comprehension of discrete-time systems, propelling progress in communication, control, and various other fields

RELATED WORK

The Cardiology team at the Arrhythmia Unit in the Hospital General University in Spain conducted a comprehensive examination of the impact of noise and distortions on Electrocardiogram (ECG) signals. They highlighted various disturbances, including baseline wander, muscle compression, and Power Line Interference (PLI), emphasizing the significance of reducing PLI noise for accurate ECG observation and management. The challenge of PLI removal was approached with a dual focus: preserving the essential ECG signal while ensuring computational efficiency for real-time tracking, particularly crucial for mobile applications receiving signals from wireless smart sensors.

In addressing this challenge, the authors aimed to develop a denoising method that is both efficient and computationally economical, considering the limited resources of mobile phones. The study explored Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters with window-based designs. Computational effectiveness and ECG signal distortion were compared, revealing that the designed IIR filter outperformed the FIR filter.

Experimental results, supported by previous research, affirmed the efficiency of FIR and IIR filters. Manpreet Kaur et al. proposed a method of removing 50Hz power line interference noise in ECG signal using the combination of moving average and IIR notch techniques (Kaur & Singh, 2009) and estimated the correlation of the reconstructed signal. Zhi-Dong Zhao et al. (2009) proposed a novel method of PLI noise reduction and baseline drift removal using machine learning mechanism. Ghorbanian et al. (2010) proposed the continuous wavelet transform method of noise reduction. The proposed method estimated the different components in the signal using neural network methods. Sorensen et al. (2010) and Alfauori

PROPOSED METHODOLOGY

To eradicate powerline interference from an Electrocardiogram (ECG) signal using a previously designed notch filter represented by the transfer function (H(z)), in conjunction with a comb filter, the following methodology was employed:

1.Frequency Analysis:
The process was initiated by scrutinizing the ECG signal to pinpoint the powerline frequency and its harmonics. Common frequencies, such as 50 Hz or 60 Hz, and their harmonics were identified within the signal.

2. Designing the Notch Filter:
The transfer function (H(z)) was utilized to formulate a specialized notch filter geared towards diminishing frequencies associated with powerline interference. Parameters (z_1) and (z_2) were selected to align with the powerline frequency and its harmonics.

3. Crafting the Comb Filter:
A comb filter was devised with a tailored transfer function to suppress various frequencies, including those of powerline interference and its harmonics. The comb filter was fashioned to complement the notch filter, collectively mitigating interference across a broader frequency spectrum.

4. Cascade Filter Design:
The notch filter and comb filter were integrated in cascade to construct a unified filter structure. This cascading ensured a synergistic operation, effectively diminishing powerline interference and its harmonics while preserving vital features of the ECG signal.

5. Implementation of the Filters:
The amalgamated notch and comb filter structure were implemented using digital signal processing techniques, specifically within the Python Spyder-IDE environment. Verification of real-time suitability was essential, particularly for continuous monitoring of ECG signals.

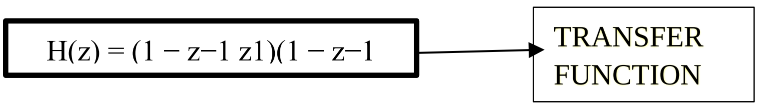
6. Adaptability Aspect:
The combined filter structure was confirmed to exhibit adaptability, enabling real-time tracking and attenuation of changing powerline frequencies. This adaptability was indispensable for accommodating variations in powerline frequencies within a constrained band.

EQUATIONS USED:

z_1=$\cos\theta + j\sin\theta$ $\qquad$ $\theta = \pm \frac{fo}{fs} \times 2\pi$

z_2= $\cos\theta - j\sin\theta$

fo=50 or 60 Hz
fs=sampling rate=1000



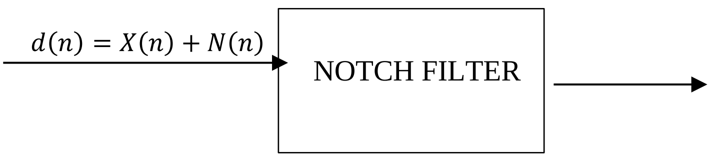


Fig1. Block diagram of Notch Filter

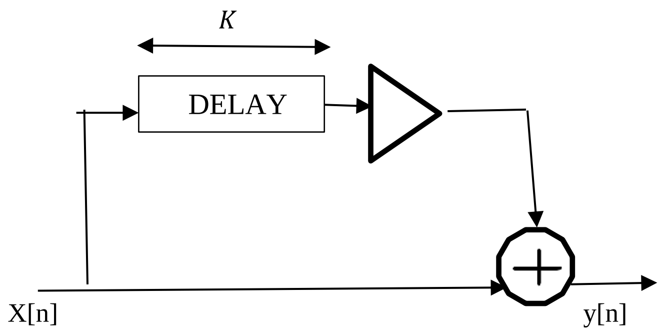
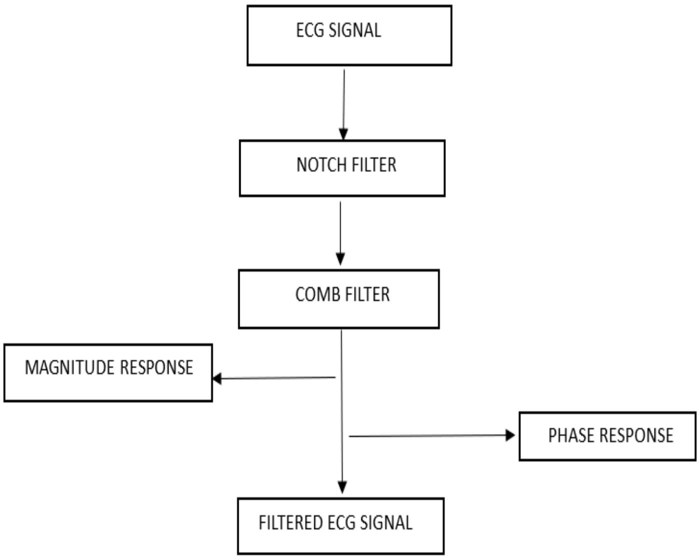


Fig2.Block diagram of Comb Filter

Fig3.Flowchat of PLI process



RESULTS AND DISCUSSIONS

A. *Dataset description:*

The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Twenty-three recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample.

The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. Two or more cardiologists independently annotated each record; disagreements were resolved to obtain the computer-readable reference annotations for each beat (approximately 110,000 annotations in all) included with the database.

This directory contains the entire MIT-BIH Arrhythmia Database. About half (25 of 48 complete records, and reference annotation files for all 48 records) of this database has been freely available here since PhysioNet's inception in September 1999. The 23 remaining signal files, which had been available only on the MIT-BIH Arrhythmia Database CD-ROM, were posted here in February 2005.

B.Performace metrics:

S.no	*Metric*	*Description*	*Value/result*
1.	*Phase linearity*	*Linearity of the phase response across frequencies*	*Visual inspection*
2.	*Cutoff Frequency*	*Freqencies at which the filter transitions*	*1.Notch filter and Comb filter:* • *Sampling rate=1000Hz* • *Magnitude response=-60,180,300Hz* • *Phase response=45-50Hz* • *Sampling rate=1500Hz* • *Magnitude response=70Hz* • *Phase response=80Hz* • *Sampling rate=500Hz* • *Magnitude response=50Hz* • *Phase response=50Hz*
3.	*Zeros*	*Distribution of poles and zeros in z-plane*	*Visual inspection*
4.	*Filtering effictivess*	*How well the filter attenuates unwanted frequencies*	*Visual inspection*

C.Simulation of Result in figures and tables:

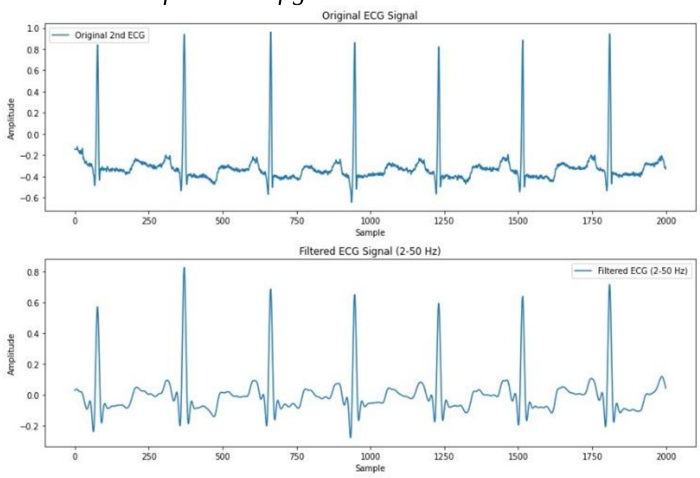


Fig4.Original ECG Signal Vs Filtered ECG Signal

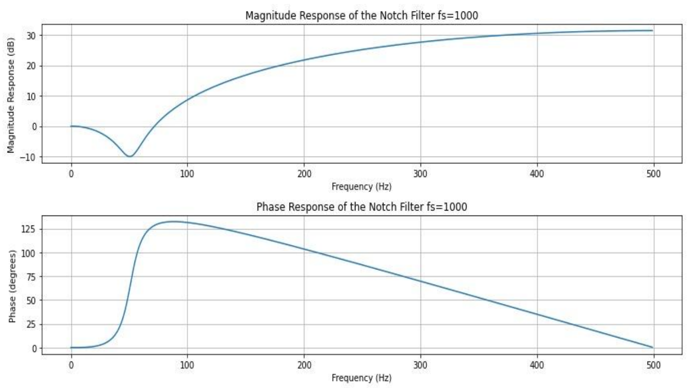


Fig5.Magnitude response & Phase response fs=1000

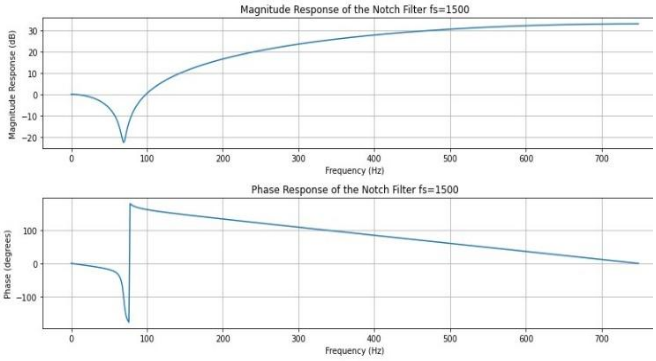


Fig6.Magnitude response & Phase response fs=1500

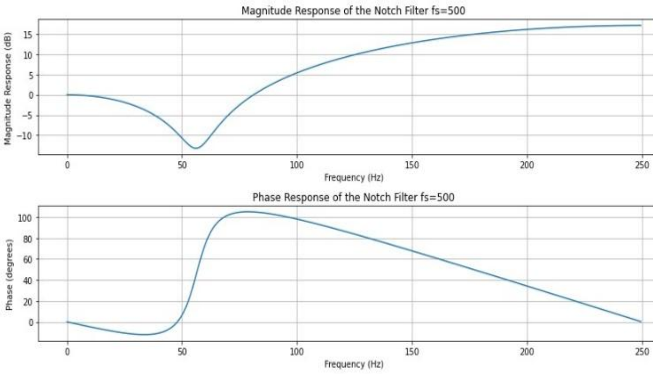


Fig7.Magnitude response & Phase response fs=500

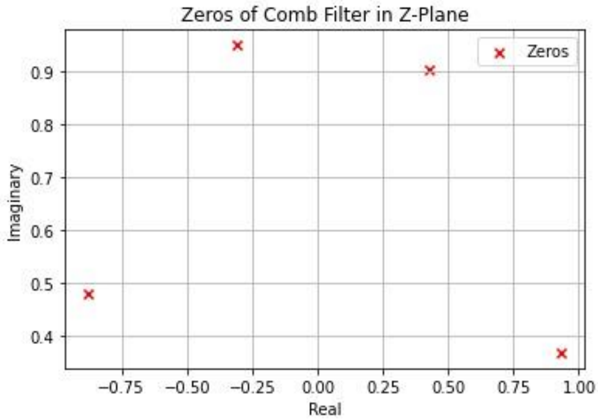


Fig8.Zeros of Comb filter

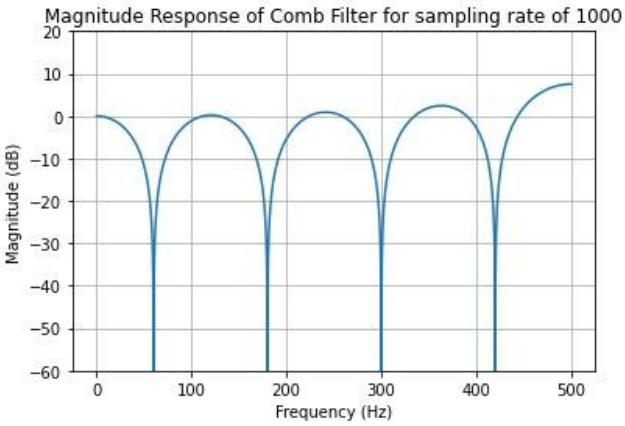


Fig9.Magnitude response of comb filter fs=1000

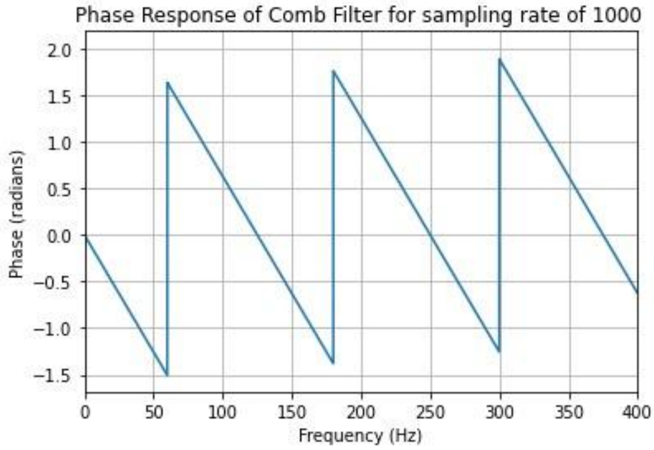


Fig10.Magnitude response of comb filter fs=1000

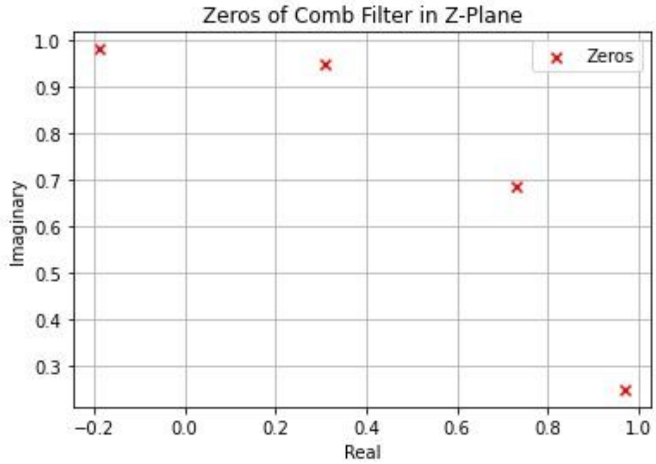


Fig11.Zeros of Comb Filter

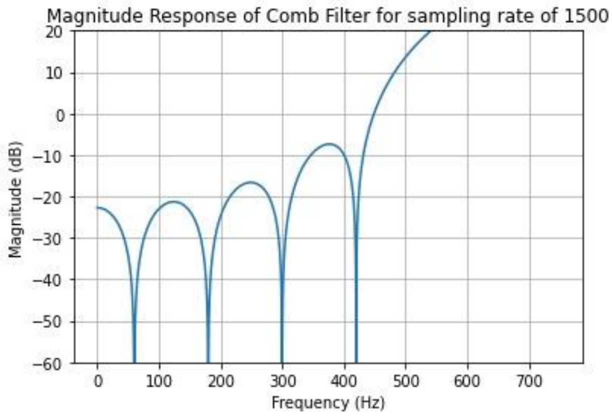


Fig12.Magnitude Response of comb filter fs=1500

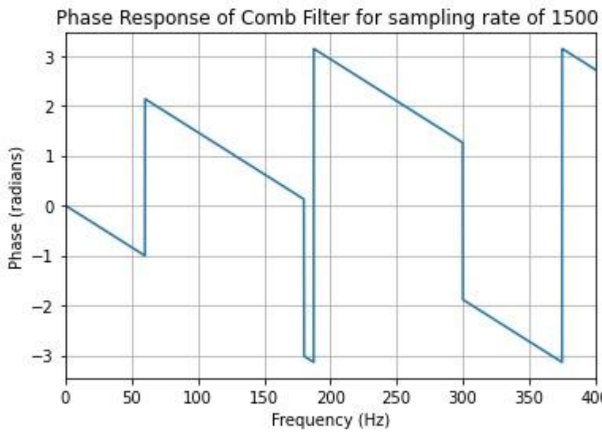


Fig13.Phase Response of comb filter fs=1500

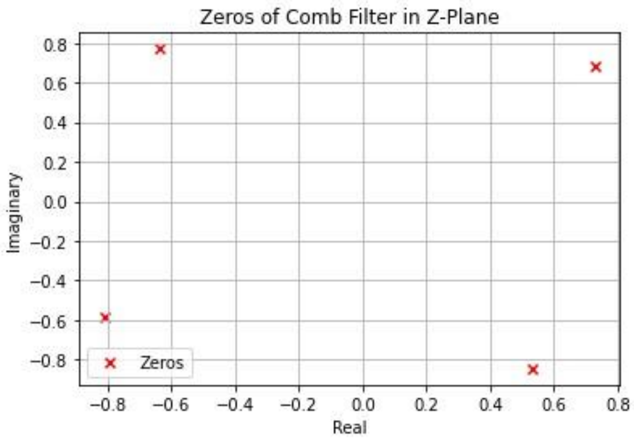


Fig14.Zeros of comb filter

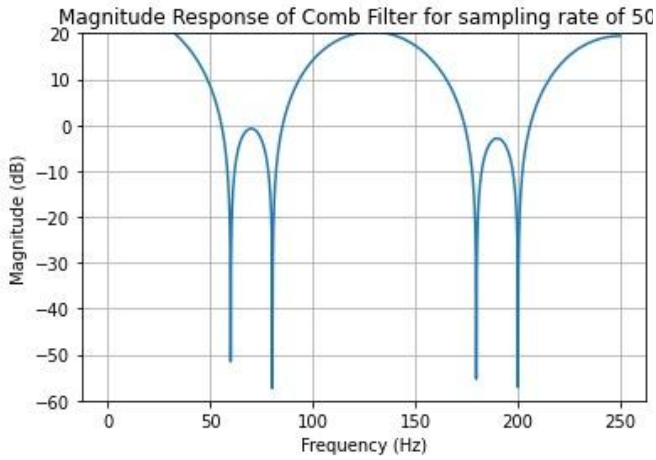


Fig15.Magnitude response of comb filter fs=500

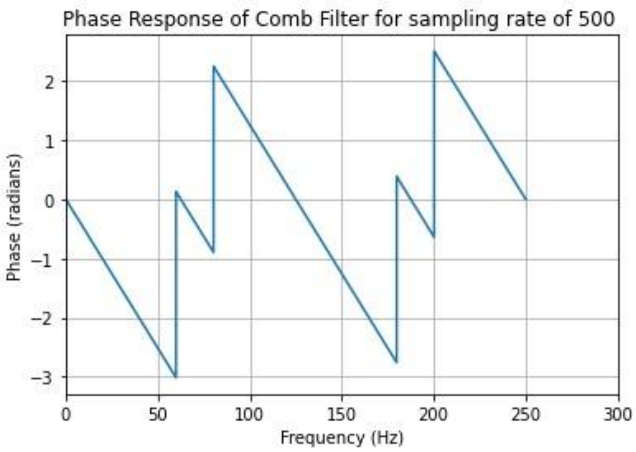


Fig16.Phase response of fs=500

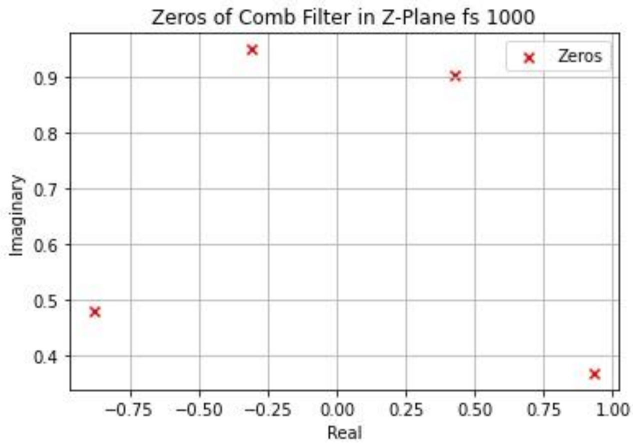


Fig17.Zeros of comb filter fs=1000

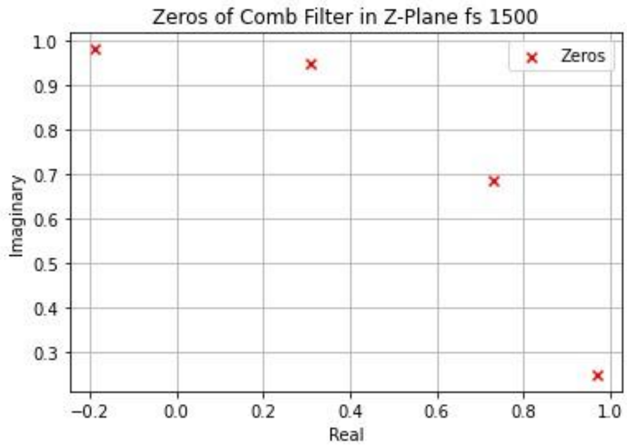


Fig18.Zeros of comb filter fs=1500

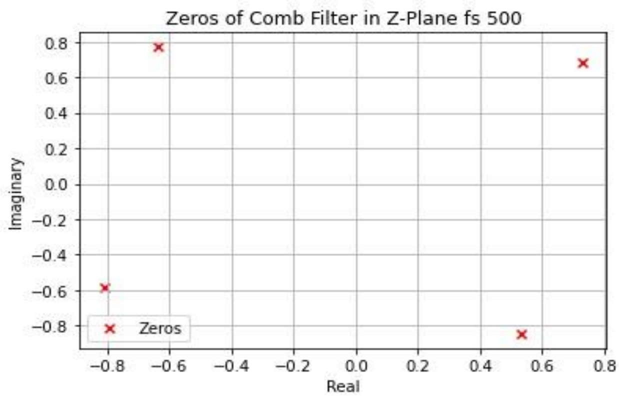


Fig19.Zeros of comb filter fs=1500

D.Inference:

When altering the sampling rate from 1000 to 1500, the transfer function equation of a notch filter becomes paramount. The transfer function of a notch filter, denoted as (H(z)), governs its frequency response. Adjusting the sampling rate necessitates a corresponding modification in the notch frequency (θ) to maintain the filter's effectiveness. The relationship between the notch frequency and the desired frequency (fo) must be recalibrated, considering the updated sampling rate (fs). The transfer function's expression, $H(z) = (1 - z^{-1} z_1)(1 - z^{-1} z_2)$, encapsulates this adjustment, where influences the filter bandwidth. Careful tuning of (θ) ensures that the notch filter retains its efficacy in suppressing unwanted frequencies, accommodating the specific requirements of the altered sampling rate for robust signal processing.

Conclusion

The electrocardiogram (ECG) report is a widely used clinical method for detecting various heart conditions;

however, it is susceptible to noise from various sources. While existing methods are limited in addressing specific types of noise with noticeable distortion after filtration, denoising remains a formidable challenge. This study primarily employs a notch and comb filter approach to achieve superior results. Our paper introduces a novel design for notch Infinite Impulse Response (IIR) filters and comb filters. The efficacy of these filters is evaluated using ECG signals as the primary testing source. We assess noise parameters, such as the correlation between the original and filtered ECG signals, observing a significant increase in output compared to input when high correlation is present. The noise parameters are identified using Python signal processing toolkits.

The results demonstrate the effectiveness, efficiency, and accuracy of the designed filter when tested with actual ECG signals and filtered ECG signals across datasets obtained from the MIT-BIH Arrhythmia Database. The correlation between the actual ECG signal and the filtered ECG is illustrated in Figure 5. Future research could extend this work by developing multiple multi-level adaptive cascaded higher-order IIR notch filter design.

REFERENCE

[1] Uzzal Biswas and Md Maniruzzaman, "Removing Power Line Interference from ECG Signal using Adaptive Filter and Notch Filter", *International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, 2014.

[2] Nauman Razzaq, "An Intelligent Adaptive Filter for Elimination of Power Line Interference From High Resolution Electrocardiogram", *IEEE Trans. date of publication*, March 2016.

[3] G. Hari Kiran and B.T. krishna, "Noise Cancellation of ECG Signal Using Adaptive Technique", *International Journal of Research in Computer and Communication Technology*, vol. 3, no. 4, April 2014S.

[4] Snehal Thalkar and Dhananjay Upasani, "Various Techniques for Removal of Power Line Interference From ECG Signal", *International Journal of Scientific & Engineering Research*, vol. 4, no. 12,December2013.Lee G.J, Yoon E.J, Kim C, Yong, K.Y, " A real-time secret image sharing with fairness,"2016.[Online].

[5] D C Reddy, Biomedical Signal Processing Principles and Techniques, pp. 198-253, 2005.

[6] M. Kaur and B. Singh, "Power Line Interference Reduction in ECG Using Combination of MA Method and IIR Notch Filter", *Int. J. of Recent Trends in Eng.*, vol. 2, no. 6, pp. 125-129, November 2009.

[7] D.G. Manolakis and V.K. Ingle, "Applied Digital Signal Processing" in , Cambrige University Press, 2011.

[8] A.A. Fedotov, "Selection of Parameters of Bandpass Filtering of the ECG Signal for Heart Rhythm Monitoring Systems", *Biomedical Engineering*, vol. 50, no. 2, pp. 114-118, Sep. 2016.

[9] E. Everss-Villalba, F.M. Melgarejo-Meseguer, M. Blanco-Velasco, F.J. Gimeno-Blanes, S. Sala-Pla, J.L. Rojo-Álvarez, et al., "Noise maps for quantitative and clinical severity towards long term ECG monitoring", *Sensors*, vol. 17, no. 11(2448), pp. 1-23, Oct. 2017.

[10] A. Milchevski and M. Gusev, "Performance evaluation of FIR and IIR filtering of ECG signals", *Proceedings of the International Conference on ICT Innovations 2016 Advances in Intelligent Systems and Computing*, vol. 665, pp. 103-112, Jan. 2018.

I. APPENDIX

Code implementation in Python:

A. Main code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import freqz, lfilter, iirnotch, butter, filtfilt
from scipy import signal
from scipy.signal import butter
import pandas as pd
def plot_comb_filter_response(zeros, sampling_rate):
  if sampling_rate==1000:
    plt.figure()
    plt.scatter(np.real(zeros), np.imag(zeros), marker='x',
color='r', label='Zeros')
    plt.title('Zeros of Comb Filter in Z-Plane fs 1000')
    plt.xlabel('Real')
    plt.ylabel('Imaginary')
    plt.legend()
    plt.grid(True)
    plt.show()
  elif sampling_rate==1500:
    plt.figure()
    plt.scatter(np.real(zeros), np.imag(zeros), marker='x',
color='r', label='Zeros')
    plt.title('Zeros of Comb Filter in Z-Plane fs 1500')
    plt.xlabel('Real')
    plt.ylabel('Imaginary')
    plt.legend()
    plt.grid(True)
    plt.show()
  elif sampling_rate==500:
    plt.figure()
    plt.scatter(np.real(zeros), np.imag(zeros), marker='x',
color='r', label='Zeros')
    plt.title('Zeros of Comb Filter in Z-Plane fs 500')
    plt.xlabel('Real')
    plt.ylabel('Imaginary')
    plt.legend()
    plt.grid(True)
    plt.show()
def comb_filter_frequency_response(b, a, sampling_rate):

    w, h = freqz(b, a, worN=8000, fs=sampling_rate)
    if sampling_rate==1000:
     plt.figure()
     plt.plot(w, 20 * np.log10(np.abs(h)))
     plt.title('Magnitude Response of Comb Filter for sampling
rate of 1000')
     plt.xlabel('Frequency (Hz)')
     plt.ylabel('Magnitude (dB)')
     plt.ylim(-60, 20)
     plt.grid(True)

     plt.show()
     plt.figure()
     plt.plot(w, np.angle(h))
     plt.title('Phase Response of Comb Filter for sampling rate of
1000 ')
     plt.xlabel('Frequency (Hz)')
     plt.ylabel('Phase (radians)')
     plt.grid(True)
     plt.xlim(0, 400)

     plt.show()
    elif sampling_rate==1500:
     plt.figure()
     plt.plot(w, 20 * np.log10(np.abs(h)))
     plt.title('Magnitude Response of Comb Filter for sampling
rate of 1500')
     plt.xlabel('Frequency (Hz)')
     plt.ylabel('Magnitude (dB)')
     plt.ylim(-60, 20)
     plt.grid(True)
```

```
    plt.show()

    plt.figure()
    plt.plot(w, np.angle(h))
    plt.title('Phase Response of Comb Filter for sampling rate
of 1500 ')
    plt.xlabel('Frequency (Hz)')
    plt.ylabel('Phase (radians)')
    plt.grid(True)
    plt.xlim(0, 400)

    plt.show()
   elif sampling_rate==500:
    plt.figure()
    plt.plot(w, 20 * np.log10(np.abs(h)))
    plt.title('Magnitude Response of Comb Filter for sampling
rate of 500')
    plt.xlabel('Frequency (Hz)')
    plt.ylabel('Magnitude (dB)')
    plt.ylim(-60, 20)
    plt.grid(True)

    plt.show()
    plt.figure()
    plt.plot(w, np.angle(h))
    plt.title('Phase Response of Comb Filter for sampling rate
of 500 ')
    plt.xlabel('Frequency (Hz)')
    plt.ylabel('Phase (radians)')
    plt.grid(True)
    plt.xlim(0, 300)

    plt.show()
def bandpass_filter(data, lowcut, highcut, fs, order=4):
   nyquist = 0.5 * fs
   low = lowcut / nyquist
   high = highcut / nyquist
   b, a = butter(order, [low, high], btype='band')


   if len(data) <= order:
      raise ValueError("The length of the input vector must be
greater than the filter order.")

   filtered_data = filtfilt(b, a, data)
   return filtered_data

csv_file_path = '100.csv'
csv_file_path_y = '100.csv'
your_sampling_rate = 1000

ecg_data = pd.read_csv(csv_file_path).iloc[0:2000,0].values
ecg_data_y =
pd.read_csv(csv_file_path_y).iloc[0:2000,0].values

lowcut = 2.0
highcut = 50.0

filtered_ecg = bandpass_filter(ecg_data, lowcut, highcut,
your_sampling_rate)
filtered_ecg_y = bandpass_filter(ecg_data_y, lowcut, highcut,
your_sampling_rate)



plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
plt.plot(ecg_data, label='Original ECG')
plt.title('Original ECG Signal')
plt.xlabel('Sample')
plt.ylabel('Amplitude')
plt.legend()


plt.subplot(2, 1, 2)
plt.plot(filtered_ecg, label='Filtered ECG (2-50 Hz)')
plt.title('Filtered ECG Signal (2-50 Hz)')
plt.xlabel('Sample')
plt.ylabel('Amplitude')
plt.legend()

plt.tight_layout()
plt.show()


plt.figure(figsize=(12, 8))

plt.subplot(2, 1, 1)
plt.plot(ecg_data_y, label='Original 2nd ECG')
plt.title('Original ECG Signal')
plt.xlabel('Sample')
plt.ylabel('Amplitude')
plt.legend()


plt.subplot(2, 1, 2)
plt.plot(filtered_ecg_y, label='Filtered ECG (2-50 Hz)')
plt.title('Filtered ECG Signal (2-50 Hz)')
plt.xlabel('Sample')
plt.ylabel('Amplitude')
plt.legend()

plt.tight_layout()
plt.show()

ecg_data = np.loadtxt('ECG.txt', delimiter=',')

noisy_ecg_data = ecg_data + 0.2 *
np.random.randn(len(ecg_data), ecg_data.shape[1])
z1=[]
z2=[]
z12=[]
z22=[]
z13=[]
z23=[]

for i in range(0,8) :
  if i%2!=0:

z1.append((np.cos((np.pi/8.33)*i)+1j*np.sin((np.pi/8.33)*i)))
    z2.append((np.cos((np.pi/8.33)*i)-1j*np.sin((np.pi/8.33)*i)))


z12.append((np.cos((np.pi/12.5)*i)+1j*np.sin((np.pi/12.5)*i)))
    z22.append((np.cos((np.pi/12.5)*i)-
1j*np.sin((np.pi/12.5)*i)))


z13.append((np.cos((np.pi/4.17)*i)+1j*np.sin((np.pi/4.17)*i)))
    z23.append((np.cos((np.pi/4.17)*i)-
1j*np.sin((np.pi/4.17)*i)))
zeros=z1
zeros1=z12
```

```
zeros2=z13
zeroes=z1
# Comb filter coefficients
b1=[0.65376569,-2.3786447,4.85001463,-
7.07101701,7.96483996, -7.07101701,4.85001463, -
2.3786447,   0.65376569]#for sampling period of 1500
b=[0.6310, -0.2149, 0.1512, -0.1288, 0.1227, -0.1288, 0.1512,
-0.2149, 0.6310] #for sampling period of 1000
# Comb filter coefficients
b2 = [3.20512821 ,1.18700923 ,1.00696344 ,1.28091747
,5.87026854, 1.28091747,1.00696344, 1.18700923,
3.20512821]#for sampling period of 500
a = [1]


def notch_filter_frequency_response(zeros,fs):
   if fs == 1000:

    b = np.poly(zeros)
    a = [1] + [0] * (len(zeros) - 1)

    b_normalized = np.array(b) / np.abs(np.polyval(b, 1))



    frequencies, response = signal.freqz(b_normalized, a, fs=fs)



    plt.figure(figsize=(12, 6))

    plt.subplot(2, 1, 1)
    plt.plot(frequencies, 20 * np.log10(np.abs(response)))
    plt.title('Magnitude Response of the Notch Filter fs=1000')
    plt.xlabel('Frequency (Hz)')
    plt.ylabel('Magnitude Response (dB)')
    plt.grid(True)

    plt.subplot(2, 1, 2)
    plt.plot(frequencies, np.angle(response, deg=True))
    plt.title('Phase Response of the Notch Filter fs=1000')
    plt.xlabel('Frequency (Hz)')
    plt.ylabel('Phase (degrees)')

    plt.grid(True)

    plt.tight_layout()
    plt.show()
   elif fs==1500:
     b = np.poly(zeros)
     a = [1] + [0] * (len(zeros) - 1)

     b_normalized = np.array(b) / np.abs(np.polyval(b, 1))

     #
     frequencies, response = signal.freqz(b_normalized, a,
fs=fs)


     plt.figure(figsize=(12, 6))

     plt.subplot(2, 1, 1)
     plt.plot(frequencies, 20 * np.log10(np.abs(response)))
     plt.title('Magnitude Response of the Notch Filter fs=1500')
     plt.xlabel('Frequency (Hz)')
     plt.ylabel('Magnitude Response (dB)')
     plt.grid(True)

     plt.subplot(2, 1, 2)
     plt.plot(frequencies, np.angle(response, deg=True))
     plt.title('Phase Response of the Notch Filter fs=1500')
     plt.xlabel('Frequency (Hz)')
     plt.ylabel('Phase (degrees)')

     plt.grid(True)

     plt.tight_layout()
     plt.show()
   elif fs==500:
     b = np.poly(zeros)
     a = [1] + [0] * (len(zeros) - 1)


     b_normalized = np.array(b) / np.abs(np.polyval(b, 1))


     frequencies, response = signal.freqz(b_normalized, a, fs=fs)


     plt.figure(figsize=(12, 6))

     plt.subplot(2, 1, 1)
     plt.plot(frequencies, 20 * np.log10(np.abs(response)))
     plt.title('Magnitude Response of the Notch Filter fs=500' )
     plt.xlabel('Frequency (Hz)')
     plt.ylabel('Magnitude Response (dB)')
     plt.grid(True)

     plt.subplot(2, 1, 2)
     plt.plot(frequencies, np.angle(response, deg=True))
     plt.title('Phase Response of the Notch Filter fs=500')
     plt.xlabel('Frequency (Hz)')
     plt.ylabel('Phase (degrees)')

     plt.grid(True)

     plt.tight_layout()
     plt.show()
fs=1000
fs1=1500
fs2=500

zero=[0.9+0.3j,0.9-0.3j]
notch_filter_frequency_response(zero,fs)
zero1=[0.9685+0.2896j,0.9685-0.2986j]
notch_filter_frequency_response(zero1,fs1)
zero2 = [0.7 + 0.6j, 0.7 - 0.6j]
notch_filter_frequency_response(zero2,fs2)

# Sampling rate
sampling_rate = 1000
sampling_rate1=1500
sampling_rate2=500

zeros=z1
zeros1=z12
zeros2=z13

plot_comb_filter_response(zeros, sampling_rate)
comb_filter_frequency_response(b, a, sampling_rate)

plot_comb_filter_response(zeros1, sampling_rate1)
comb_filter_frequency_response(b1, a, sampling_rate1)

plot_comb_filter_response(zeros2, sampling_rate2)
comb_filter_frequency_response(b2, a, sampling_rate2)
```