WIENER FILTERS FOR PREDICTION AND FILTERING

Project report

Submitted by:

22BEC025: Aayush Desai

Semester – 4

Batch – A2

Submitted to:

Faculty: Sudhanshu Janwadkar

2EC402 - Digital Signal Processing

Department of Electronics and Communication Engineering

School of Technology

Nirma University

INTRODUCTION:

In signal processing, the Wiener filter is a filter used to produce an estimate of a desired or target random process by linear time-invariant (LTI) filtering of an observed noisy process, assuming known stationary signal and noise spectra, and additive noise. The Wiener filter minimizes the mean square error between the estimated random process and the desired process.

One of the first techniques developed to reduce the effect of additive random noise in images is the so-called Wiener filtering. Wiener's filtering is based on the assumption that additive noise is a stationary random process, which means it is independent of the pixel location. The window dimension is raised, and the variable cutoff frequency obtains with lower cutoffs in areas with fewer features and higher cutoffs in regions with edges or other high-variance characteristics. The overall frequency response is dependent on the window size: the larger the window, the more the cutoff is lowered, increasing the overall blurring and noise reduction. Wiener filter uses filtering of noisy signal using the spectral properties of desired signal and noise present considering both as stochastic process with linear property.

The Wiener filter is broadly utilized in many areas, which gets image deblurring or signal de-noising and thus made image sharpness and remove noise to accomplish a variety of applications ranging from cleaning medical images, filtering speech signals, filtering remote sensing images. In addition to the above structure, Wiener filters perform and provide excellent results, the Wiener filter helps to improve target congestion in radar and sonar systems, refining acquired data from geophysical exploration to refine seismic imaging, and optimizing imaging in optical systems such as large telescopes and microscopes . Wiener filters are also used to enhance the quality of video followed by surveillance, broadcasting, and video conferencing tasks.

EQUATIONS AND BLOCK DIAGRAM:

- This linear filter is applied with coefficients WK on the estimated signal. The input signal, d(n) consist of noise, v(n),

$$x(n) = d(n) + v(n)$$

- The output signal, y (n) should be a close estimate of d (n). So the error signal e (n) should be minimum. The adaptive algorithm tries to correct the weights W_k, so that the mean square error is minimized.

$$err = \min(E(e(n)^2)$$

where, $e(n) = y(n) – d(n)$

- A k tap discrete Wiener filter uses the following equation to find the value of y (n)

$$\mathrm{y(n)} = \sum_{k=0}^{N-1} W(k)\ (\mathrm{d(n-k)} * \mathrm{v(n-k)})$$

- The Wiener-Hopf equation, which calculates the optimal weights, is the most significant feature of the Wiener filter.

$$\mathrm{rxd}(-\mathrm{l}) = \sum_{l=0}^{p-1} W'^{(l)} \mathrm{rxx(k-1)}$$

where W' are the optimum values of tap weights of the filter and rxx is the autocorrelation function of noise and rxd is the cross correlation function between noise and d(n).

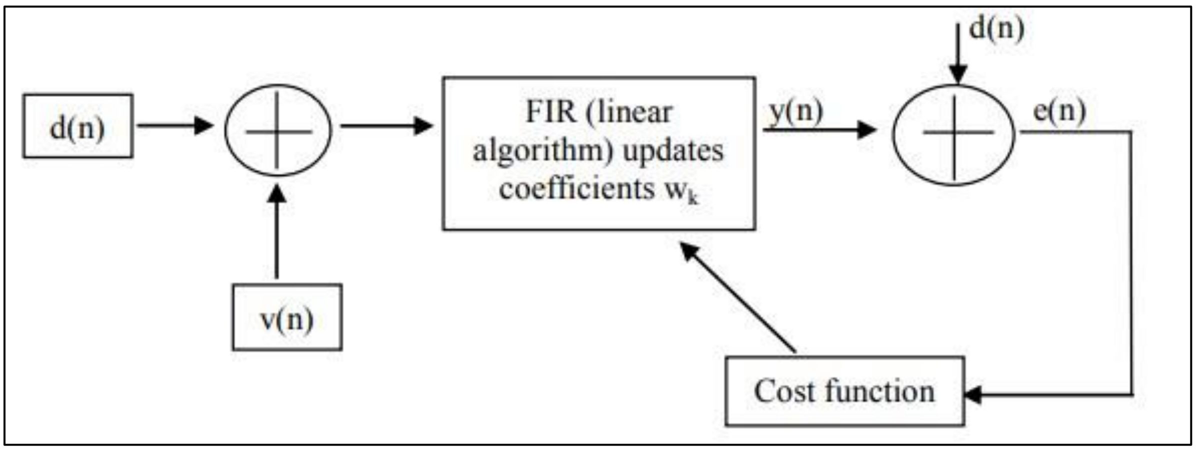


Figure: Block Diagram of wiener filter.

ASSUMPTIONS:

The Wiener filter relies on several key assumptions for its estimations and effectiveness. These assumptions include:

1. Stationarity: Signal and noise have constant statistical properties.
2. Linearity: The system is linear, outputting a combination of input signal and noise.
3. Additivity: Observed signal is the sum of desired signal and noise.
4. Gaussian noise: Noise follows a Gaussian distribution.
5. Knowledge requirement: Wiener filter needs signal and noise statistics knowledge.
6. Signal-to-noise ratio (SNR) requirement: Wiener filter needs SNR knowledge for trade-off determination.
7. Time-invariance: System generating the signal is time-invariant.

METHODOLOGY:

I. Wiener Filter for removing noise from an image.

The Wiener filter is effective in reducing noise from signals such as audio signals, biomedical signals, and communication signals. By adaptively adjusting filter coefficients based on signal characteristics, it can effectively suppress noise while preserving signal features.

Given below is the code to filter an image using MATLAB function **wiener2()**.

```
% WEINER FILTER FOR IMAGE PROCESSING (USING WEINER2() FUNCTION)

% Read the image to process
RGB = imread('flower.jpg');
figure
imshow(RGB);
title('3-D coloured image');

% Convert the image to grey scale
I = im2gray(RGB);
figure
imshow(I);
title('Image in gray scale');

% Add Gaussian noise to image
J = imnoise(I,'gaussian',0,0.025);
figure
imshow(J);
title('Image with Added Gaussian Noise');

% Pass the noise added image through wiener filter
K = wiener2(J,[5 5]);
figure
imshow(K);
title('Image with Noise Removed by Wiener Filter');
```

Functions used:

- **imread()** reads the image from the file specified by filename, inferring the format of the file from its contents.

  RGB = imread(filename)

- **imshow()** display RGB image or grayscale image

  Imshow(image_name)

- **im2gray()** converts the specified truecolor image RGB to a grayscale intensity image I.

  I = im2gray(RGB)

- **imnoise()** adds Gaussian white noise with mean m and variance var_gauss.

  J = imnoise(I,'gaussian',m,var_gauss)

- **wiener2()** filters the grayscale image I using a pixel-wise adaptive low-pass Wiener filter.

  K = wiener2(I,[m n])

  [m n] specifies the size (m-by-n) of the neighborhood used to estimate the local image mean and standard deviation. The additive noise (Gaussian white noise) power is assumed to be noise.

Results:

(a.) (b.) (c.) (d.)

Figure: (a.) RGB image. (b.) image in grayscale (c.) Image with added gaussian noise (d.) Image with noise removed by wiener filter.

II. Wiener Filter for de-blurring an image.

One of the primary applications of the Wiener filter is in image de-blurring. It can be used to remove blur caused by motion, defocus, or other factors, restoring sharpness and clarity to images.

Given below is the code to deblur an image using MATLAB function **deconvwnr()**.

```
% WEINER FILTER ALGORITHM TO DEBLUR THE IMAGE USING deconvwnr() FUNCTION

% Read the blurry image
blurry_image = imread('blur.jpg');

% Display the blurry image
figure;
subplot(1, 2, 1);
imshow(blurry_image);
title('Blurry Image');

% Define the PSF (Point Spread Function) for the blurring process
psf = fspecial('gaussian', [5 5], 2);

% Set the parameters for deblurring
lambda = 0.01; % Regularization parameter (for Wiener filter)

% Deblur the image using the Wiener filter
deblurred_image_wiener = deconvwnr(blurry_image, psf, lambda);

% Display the deblurred image using the Wiener filter
subplot(1, 2, 2);
imshow(deblurred_image_wiener);
title('Deblurred Image');
```

Functions used:

- **fspecial()** returns a rotationally symmetric Gaussian lowpass filter of size hsize with standard deviation sigma.

psf = fspecial('gaussian',hsize,sigma)

- **deconvwnr()** deconvolves image I using the Wiener filter algorithm, returning deblurred image J.

J = deconvwnr(I,psf,nsr)

psf is the point-spread function (PSF) with which I was convolved. nsr is the noise-to-signal power ratio of the additive noise. The algorithm is optimal in a sense of least mean square error between the estimated and the true images.

Results:

(a.)

(b.)

Figure: (a.) blur image. (b.) de-blurred image.

III. Wiener Filter for removing noise from an audio signal.

Although wiener filters are widely used in the field of image processing (2-D signals), but they can also be used to filter out noise from the audio/speech signals (1-D signals). It uses the exact same algorithm named **'least mean square error method',** where the filter coefficients are optimized such that the error between the desired signal and the output is minimized to a threshold.

Equation used:

Mathematically, the Wiener filter aims to find the optimal linear filter **H(f)** that minimizes the mean square error between the estimated signal **Y(f)** and the original signal **X(f)**:

$$H(f) = \frac{S(f)}{S(f) + N(f)}$$

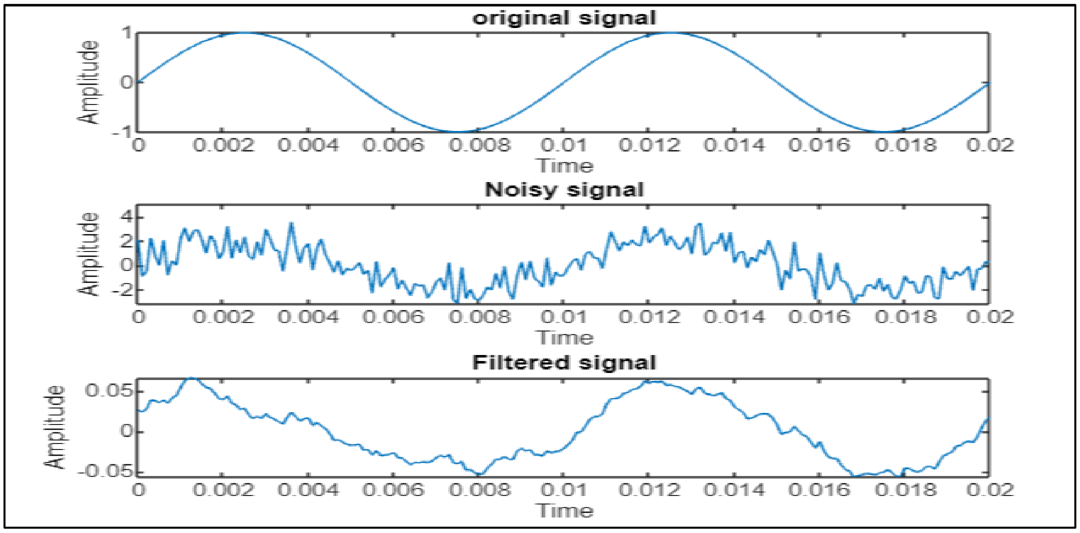
where, **S(f)** is power spectral density of the signal and **N(f)** is power spectral density of noise.

Given below is the code to filter an sinusoidal signal of low frequency using wiener filter implemented using equations:

```
% WEINER FILTER FOR AUDIO PROCESSING (using equations)

f=100;                          % frequency of input signal
sample_rate= 10000;             % sampling frequency
T= 0.02;                        % duration of signal
t= 0: 1/sample_rate: 0.02;      % sampling intervals
d= sin(2*pi*f*t);               % Original signal
noise= randn(size(d));          % Generate noise using randn() function
snr= 10^(0.5);                  % snr= 5db
noisy_audio= sqrt(snr)*d + noise; % Adding noise to original signal

S = fft(noisy_audio);           % represent noise added signal in frequency-domain

% Plot Original signal
subplot(3,1,1);
plot(t,d);
title('original signal');
xlabel('Amplitude');
ylabel('time');

% Plot noisy signal
subplot(3,1,2);
plot(t, noisy_audio);
title('Noisy signal');
xlabel('Amplitude');
ylabel('time');
```

```
% PSD of input signal
Rs= abs(fft(d)).^2/ length(d);

% PSD of noise
Rn= abs(fft(noise)).^2/ length(noise);

% PSD between input signal and noise
Rsn= abs(fft(d) .* conj(fft(noise)))/ length(d);

% Estimation of coefficients of wiener filter
H = conj(Rsn) ./ (Rs + Rn);

% Obtaining filtered signal
y= real(ifft(H .* fft(noise)));
subplot(3,1,3);
plot(t, y);
title('Filtered signal');
xlabel('Amplitude');
ylabel('time');
```

Results:



Algorithm:

1. **Initialization**: The Wiener filter requires knowledge of the signal and noise statistics, specifically the autocorrelation functions or power spectral densities of the desired signal and the noise. These statistics are typically estimated from the observed signal and noise or known a priori.

2. **Frequency Domain Representation**: Compute the Fourier transforms of the observed signal, the desired signal, and the noise. This transforms the signals from the time domain to the frequency domain, allowing for easier manipulation and filtering.

3. **Wiener-Hopf Equation**: Calculate the filter coefficients using the Wiener-Hopf equation, which optimally minimizes the mean square error between the desired signal and the filtered output. The Wiener filter coefficients $H(f)$ in the frequency domain are given by:

$$H(f) = \frac{Rsd(f)}{Rss(f) + Rnn(f)}$$

   where:

   - $Rss(f)$ is the power spectral density (PSD) of the desired signal.
   - $Rnn(f)$ is the PSD of the noise.
   - $Rsd(f)$ is the cross-power spectral density (CPSD) between the desired signal and the observed signal.

4. **Filtering**: Multiply the Fourier transform of the observed signal by the Wiener filter coefficients $H(f)$ to obtain the filtered signal in the frequency domain.

5. **Inverse Fourier Transform**: Apply the inverse Fourier transform to the filtered signal to obtain the time-domain representation of the enhanced signal, which is the output.

APPLICATIONS:

The Wiener filter is a versatile tool used in various fields for signal processing and image processing tasks. Some of its applications include:

1. **Image Deblurring**: One of the primary applications of the Wiener filter is in image deblurring. It can be used to remove blur caused by motion, defocus, or other factors, restoring sharpness and clarity to images.
2. **Signal Denoising**: The Wiener filter is effective in reducing noise from signals such as audio signals, biomedical signals, and communication signals. By adaptively adjusting filter coefficients based on signal characteristics, it can effectively suppress noise while preserving signal features.
3. **Image Restoration**: In addition to deblurring, the Wiener filter is used for image restoration tasks such as inpainting, where missing or damaged parts of an image are reconstructed based on surrounding information.
4. **Speech Enhancement**: Wiener filtering is commonly used in speech processing applications to improve the quality of speech signals by reducing background noise while preserving speech intelligibility.
5. **Medical Imaging**: In medical imaging, the Wiener filter is applied to enhance the quality of images obtained from various modalities such as MRI, CT, ultrasound, and microscopy. It helps improve image contrast and sharpness while reducing artifacts and noise.
6. **Radar and Sonar Signal Processing**: Wiener filtering is used in radar and sonar systems for target detection and tracking. It helps improve the accuracy of target identification by enhancing signal-to-noise ratio and reducing clutter.
7. **Seismic Data Processing**: In geophysical exploration, the Wiener filter is applied to process seismic data for oil and gas exploration, earthquake detection, and subsurface imaging. It helps improve the resolution and clarity of seismic images.
8. **Remote Sensing**: Wiener filtering is utilized in remote sensing applications for satellite and aerial imagery processing. It helps remove noise and atmospheric effects, allowing for accurate analysis and interpretation of remote sensing data.
9. **Optical Imaging**: In optical imaging systems such as telescopes and microscopes, the Wiener filter is employed to improve image quality by compensating for optical aberrations and noise.
10. **Video Enhancement**: Wiener filtering can be applied to enhance the quality of video sequences by reducing noise and improving sharpness and detail. It is used in video surveillance, broadcasting, and video conferencing applications.

LIMITATIONS:

Limitations of the Wiener filter include:

1. **Sensitivity to Model Assumptions**: Wiener filter performance is affected by assumptions like linearity and stationarity.
2. **Requirement for Signal and Noise Statistics**: Accurate estimation of signal and noise characteristics is crucial for effective filtering.
3. **Sensitivity to Signal-to-Noise Ratio (SNR)**: Wiener filter effectiveness varies with SNR, performing poorly at low SNR levels.
4. **Trade-off between Noise Reduction and Signal Distortion**: Balancing noise reduction with signal preservation is challenging, especially in complex signals.
5. **Computational Complexity**: Wiener filter implementations can be resource-intensive, posing challenges for real-time applications.
6. **Limited Effectiveness for Non-Gaussian Noise**: Wiener filter performance may degrade when noise does not follow a Gaussian distribution.
7. **Lack of Adaptability to Dynamic Environments**: Wiener filter assumes stationary conditions, limiting its effectiveness in changing environments.
8. **Lack of Robustness to Model Mismatch**: Inaccurate assumptions or estimations can lead to suboptimal filtering results.

RECENT ADVANCES AND FUTURE SCOPES:

Recent advances in Wiener filters include integration of deep learning, adaptive filtering, sparse sensing, and Bayesian inference techniques, along with multi-scale filtering and real-time implementation for improved adaptability, efficiency, and performance across various applications.

Future directions focus on further enhancing adaptability in dynamic environments, robustness to uncertainties, and applications in emerging technologies such as IoT, autonomous systems, and VR/AR.

CONCLUSION:

In conclusion, this report has provided a comprehensive overview of Wiener filters, covering their principles, applications, limitations, recent advances, and future directions. Wiener filters play a crucial role in signal processing tasks, offering effective solutions for noise reduction, signal enhancement, and restoration in diverse domains such as image processing, audio processing, communications, biomedical signal processing, and remote sensing. Despite their effectiveness, Wiener filters have limitations, including sensitivity to model assumptions, requirements for accurate signal and noise statistics, and challenges in dynamic environments.

REFERENCES:

1. ECG Denoising using Wiener filter and Kalman filter (2020) - Manju B. R. and Sneha M. R.
   https://www.sciencedirect.com/science/article/pii/S1877050920309947
2. Noise reduction by Wiener filter – Yi wen chen
   https://youtu.be/ATnP3c-NANI?si=POjwCS2ZnckQ4P4Y
3. Wiener(Minimum Mean Square Error)Filter in Digital Image Processing and its implementation in MATLAB
   https://youtu.be/hiCxRTNhdHw?si=4z6_RJnxatF8j0Tx
4. Weiner Filter – Wikipedia
   https://en.wikipedia.org/wiki/Wiener_filter
5. 2-D adaptive noise-removal filtering – MATLAB wiener2
   https://www.mathworks.com/help/images/ref/wiener2.html
6. Deblur image using Wiener filter – MATLAB deconvwnr
   https://www.mathworks.com/help/images/ref/deconvwnr.html
7. The Wiener filter – The University of Edinburgh
   https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/VELDHUIZEN/node15.html
8. Wiener filters – Slideshare (Chapter 2 wiener filters from Adaptive filter theory simon haykin)
   https://www.slideshare.net/fsfa786/chapter2-wiener-filters