

MATLAB Audio Noise Removal Activity Guide

Digital Signal Processing Laboratory

In this activity, you will record your own voice in MATLAB, add high-frequency noise and develop an FIR low-pass filter to recover the recording. Your task is to use evidence from time-domain plots, frequency spectra and audio playback to select, test and justify an effective filter design.

Student name(s)	
Student ID(s)	
Class or group	
Date	

Activity at a Glance

Item	Details
Activity type	Problem-based MATLAB laboratory
Suggested duration	Approximately 3 hours
Working arrangement	Individual or pairs
Software	MATLAB with Signal Processing Toolbox
Required output	MATLAB script, labelled figures, results table and technical discussion

Problem Scenario

A voice recording required for a communication or monitoring application has been contaminated by unwanted high-frequency noise. The noise reduces the clarity of the recording and makes the signal difficult to interpret. You have been assigned to record your own voice in MATLAB, deliberately add a controlled high-frequency noise component, and develop a digital filter that reduces the noise while preserving as much useful speech information as possible.

Your final filter must be supported by evidence. You must identify the unwanted frequency content, select appropriate filter specifications, compare the signal before and after filtering, and explain any limitations or trade-offs in your design.

Aim

To design and evaluate a MATLAB-based FIR low-pass filter for reducing high-frequency noise in an audio signal.

Learning Objectives

At the end of this activity, you should be able to:

- Record, retrieve, inspect and play a personal voice signal in MATLAB.
- Represent an audio signal in the time and frequency domains.
- Use the Fast Fourier Transform to identify useful and unwanted frequency components.
- Generate and add a controlled high-frequency noise signal to an audio recording.
- Select appropriate FIR low-pass filter specifications based on signal evidence.
- Design and apply an FIR filter using MATLAB.
- Compare filter designs and evaluate the effect of filter order, cutoff frequency and window type.
- Interpret graphical and audible results and justify a recommended filter design.

Background Information

Audio Signals and Sampling

A digital audio signal is a sequence of samples representing the amplitude of a sound waveform at equally spaced times. The sampling frequency, F_s , states how many samples are collected each second. According to the Nyquist principle, the highest frequency that can be represented without aliasing is $F_s/2$.

Time and Frequency Domains

A time-domain plot shows how signal amplitude changes with time. A frequency-domain plot shows how the signal is distributed across frequency. The FFT is an efficient method of calculating the discrete Fourier transform and is used in this activity to locate dominant audio and noise frequencies.

FIR Low-Pass Filtering

A low-pass filter allows frequencies below its cutoff region to pass while reducing higher frequencies. A finite impulse response filter uses a finite set of coefficients and can be designed to have linear phase. Its performance depends on factors including filter order, cutoff frequency and the window applied during design. A higher order may produce a sharper transition but increases computation and delay.

Required Resources

Resource	Purpose
Computer with MATLAB	Programming, analysis, plotting, filter design and audio playback.
Signal Processing Toolbox	Provides functions used for FIR design and signal analysis.
Working microphone	Allows each student to record a personal voice sample directly in MATLAB.
Headphones	Allows careful comparison of clean, noisy and filtered audio

Resource	Purpose
	without disturbing others.
Starter MATLAB script	Provides the required file structure and selected setup commands.
This activity guide	Records the procedure, results, analysis and submission requirements.

Responsible Audio Practice

- Use headphones at a low, comfortable volume. Reduce the computer volume before the first playback.
- Use only your own voice and do not state private or personally sensitive information in the recording.
- Save your MATLAB script frequently and retain an unchanged copy of your original voice recording.
- Use clear filenames and keep all activity files in one working folder.

Pre-Lab Preparation

Complete the following before beginning the practical work:

- Review sampling frequency, Nyquist frequency and aliasing.
- Review the purpose of the FFT and the interpretation of a magnitude spectrum.
- Review the function of a low-pass filter and the meaning of cutoff frequency and filter order.
- Confirm that MATLAB has permission to access the microphone and that audio recording and playback are working.

P1. If the audio sampling frequency is 44,100 Hz, calculate the Nyquist frequency.

P2. Explain why a low-pass filter may be appropriate for removing high-frequency noise from speech.

P3. State one possible disadvantage of selecting a cutoff frequency that is too low.

Activity Procedure

Part 1 Record and Inspect Your Voice

Use MATLAB to record approximately five to ten seconds of your own voice. Speak clearly, maintain a consistent distance from the microphone and record in a reasonably quiet location. Use a sampling frequency approved by your lecturer and save an unchanged copy of the recording for later comparison.

```
recObj = audiorecorder(Fs, 16, 1);  
recordblocking(recObj, recordingDuration);  
voice = getaudiodata(recObj);
```

Create a time vector, play the signal at a safe volume, and plot amplitude against time. Your figure must include a title, axis labels and a grid. Export the original voice recording as a WAV file if instructed.

- Record the sampling frequency and number of samples.
- Calculate the recording duration.
- Record the selected bit depth and number of audio channels.
- Describe any visible features of the time-domain waveform.

Part 2 Analyse the Original Frequency Content

Use the FFT to calculate the frequency spectrum of the original signal. Construct an appropriate frequency axis and plot a single-sided magnitude spectrum. Limit the displayed frequency range so that the important audio components can be interpreted clearly.

Identify the main frequency range occupied by the useful audio and record any dominant spectral peaks. Do not select a cutoff frequency until you have examined this evidence.

Part 3 Generate the Noisy Audio Signal

Generate a sinusoidal noise signal at a frequency above the main useful audio range. Select an amplitude that makes the noise noticeable without causing clipping. Add the noise to the original audio and normalise the result if required.

Listen to the noisy signal and produce both time-domain and frequency-domain plots. Confirm that the added noise produces a distinct spectral component at the expected frequency.

Part 4 Select the Filter Specifications

Use your frequency-domain evidence to propose an FIR low-pass filter. Record the initial specifications in the table below before writing the filter-design section of your code.

Parameter	Selected value	Reason for selection
Sampling frequency		
Estimated useful frequency		

Parameter	Selected value	Reason for selection
range		
Noise frequency		
Cutoff frequency		
Filter order		
Window type		

Part 5 Design and Apply the FIR Filter

Design an FIR low-pass filter using your selected order, cutoff frequency and window. Apply the filter to the noisy audio signal. Plot the filter frequency response and verify that the noise frequency falls within the attenuated region.

Listen to the filtered signal. If the useful audio is distorted or the noise remains excessive, revise one parameter at a time and repeat the test. Keep a record of each meaningful trial.

Part 6 Compare Alternative Designs

Test at least three filter configurations. At least one parameter must change between trials. You may compare different filter orders, cutoff frequencies or windowing methods. Use the same noisy input signal for every trial so that the comparison is fair.

Trial	Order	Cutoff frequency	Window	Observed noise reduction	Effect on useful audio
1					
2					
3					

Part 7 Present the Final Comparison

Prepare a final set of clearly labelled figures showing:

- The original voice waveform and frequency spectrum.
- The noisy audio waveform and frequency spectrum.
- The filtered audio waveform and frequency spectrum.
- The frequency response of the selected FIR filter.

Use consistent axes where appropriate so that differences can be compared fairly. Every figure must have a descriptive title, labelled axes, units and a legend when more than one signal is shown.

Required MATLAB Tasks

Task	Minimum requirement	Evidence to submit
Voice recording	Record and retrieve a five- to ten-second personal voice sample in MATLAB.	Code and recording information.
Time analysis	Create a correct time vector and plot the waveform.	Labelled time-domain figure.
FFT analysis	Calculate and plot a single-sided spectrum.	Labelled frequency-domain figure.
Noise generation	Add controlled high-frequency noise without clipping.	Code, selected values and noisy-signal plots.
Filter design	Design an FIR low-pass filter from justified specifications.	Specifications and filter-response plot.
Filter application	Process the noisy signal and play the output.	Code and filtered-signal plots.
Design comparison	Test at least three configurations.	Completed trial table and discussion.
Final recommendation	Select and justify the best design.	Evidence-based conclusion.

Results to Record

Result	Value or observation
Sampling frequency	
Voice-recording duration	
Bit depth and number of channels	
Nyquist frequency	
Selected noise frequency and amplitude	

Result	Value or observation
Dominant useful frequency range	
Final cutoff frequency	
Final filter order	
Final window type	
Overall effect of filtering	

Analysis and Discussion Questions

A1. Compare the original and noisy frequency spectra. What evidence confirms the frequency and relative strength of the added noise?

A2. Explain how you selected the cutoff frequency. Refer to the useful audio range and the noise frequency in your response.

A3. How did increasing or decreasing the filter order affect the transition between the passband and stopband? What trade-off did you observe?

A4. Compare the windowing methods tested. Which produced the most suitable result for this signal, and why?

A5. Compare the filtered signal with the original signal in both the time and frequency domains. What useful information, if any, was reduced by the filter?

A6. Was audio playback alone sufficient to determine the best filter? Explain why graphical or numerical evidence was also necessary.

A7. Identify one limitation of using a low-pass filter for noise removal. Describe a situation in which this method would be unsuitable.

A8. Recommend your final filter design. Support your recommendation using the filter specifications, plots and listening observations.

Conclusion

Write a concise conclusion stating whether the activity aim was achieved. Identify the selected filter, summarise the main evidence supporting its effectiveness and state one limitation of the final design.

Submission Requirements

Submit the following items as instructed by your lecturer:

- One clearly organised and commented MATLAB script.
- The completed activity guide or technical report.
- All required labelled figures in a readable format.
- The completed filter-comparison and results tables.
- Responses to all analysis and discussion questions.
- A concise conclusion and recommendation.
- The original voice recording and filtered audio file, if requested.

File Naming

Use a clear naming convention such as:

StudentID_AudioNoiseRemoval.m

StudentID_AudioNoiseRemoval_Report.docx

StudentID_OriginalVoice.wav

StudentID_FilteredAudio.wav

Quality Checklist

- ☐ The MATLAB script runs from beginning to end without unresolved errors.
- ☐ All selected filter parameters are defined and explained.
- ☐ The original, noisy and filtered signals are compared fairly.
- ☐ Every figure has a title, axis labels, units and a legend where necessary.
- ☐ The discussion refers to specific evidence from the results.
- ☐ The original voice, filtered audio and document files use the required filenames.