Acoustic Roughness EE 399 Capstone Project

Cristian Hernandez

Northwestern University ECE

Acoustic Roughness EE 399 Capstone Project

by

Cristian Hernandez

Student:	Cristian Hernandez
Professor:	Thrasyvoulos Pappas
Project Duration:	MARCH 2024 - JUNE 2024
University Affiliation:	Northwestern University McCormick School of Engineering and Applied Sciences

Preface

This report is the culmination of extensive research and development undertaken as part of an engineering capstone project aimed at exploring the complex phenomenon of acoustic roughness. Acoustic roughness, characterized by the perceptual attribute of rapid fluctuations or harshness in sound, plays a significant role in various auditory experiences. Understanding this concept is crucial for advancements in fields such as audio signal processing, music synthesis, and auditory research. Through the comprehensive study and analysis of two seminal research papers, coupled with the development of an advanced MATLAB simulation, this report seeks to deepen our understanding of acoustic roughness and its implications. The interactive MATLAB simulation developed as part of this project provides a powerful tool for visualizing and analyzing the effects of various parameters on roughness, offering valuable insights for both researchers and practitioners in the field. This preface sets the stage for a detailed exploration of acoustic roughness, highlighting the importance of this study and the innovative approaches employed to achieve its objectives.

Cristian Hernandez
Northwestern University, June 2024

Contents

1 Introduction

Acoustic roughness, an auditory attribute marked by rapid fluctuations or harshness in sound, is a critical aspect of our auditory perception. This phenomenon is often linked to the modulation of amplitude or frequency within a certain range and plays a pivotal role in the perception of complex sounds, such as musical timbres and environmental noises. Acoustic roughness is not just a theoretical concept but has practical implications in various fields including audio signal processing, music synthesis, auditory research, and hearing aid design. Understanding how roughness is perceived and quantified can lead to significant advancements in these areas, enhancing our ability to manipulate and improve sound quality. This report explores acoustic roughness by summarizing key findings from two foundational research papers and detailing the creation of an advanced MATLAB simulation designed to visualize and analyze roughness. The simulation allows for real-time manipulation of sound parameters, providing a hands-on approach to studying the intricate relationships between spectral and temporal characteristics of sound and their effects on perceived roughness. Through this comprehensive study, we aim to contribute to the broader understanding of acoustic roughness and its practical applications in various auditory-related fields.

2

Review of "Two-Phase Effects in Roughness Perception" by Daniel Pressnitzer and Stephen McAdams

The seminal paper ”Two-Phase Effects in Roughness Perception” by Daniel Pressnitzer and Stephen McAdams provides a detailed exploration of how spectral and temporal characteristics of sound interact to influence the perception of acoustic roughness. The authors embark on a rigorous investigation into the perceptual effects of phase relationships among sound components and the shape of sound envelopes, revealing the nuanced ways in which these factors contribute to the experience of roughness.

At the core of their study, Pressnitzer and McAdams examine the role of phase variation in the perception of roughness. By manipulating the phase of the central component in a three-component signal, they demonstrate that changes in phase relationships significantly alter the temporal structure of the sound, thereby affecting perceived roughness. This finding highlights the critical importance of phase information in auditory processing, as the auditory system does not only process the frequency content of sounds but also the precise timing and phase relationships between components. The researchers employed a methodical approach by systematically varying the phase of a sinusoidal component relative to two others, allowing them to observe the resultant changes in auditory perception and quantify the impact of these variations on roughness.

The study further delves into the effects of envelope shape on roughness perception. Using sawtooth modulation to create asymmetrical envelopes, Pressnitzer and McAdams illustrate that the shape of the envelope profoundly influences how roughness is perceived. This aspect of their research involved presenting participants with sounds that had identical spectral content but different temporal modulations. The findings revealed that sounds with more pronounced asymmetrical envelopes were perceived as rougher compared to those with symmetrical or less pronounced envelopes. This result underscores the interplay between temporal asymmetry and auditory perception, suggesting that our auditory system is highly sensitive to the temporal structure of amplitude variations in sound. The ability to distinguish between different envelope shapes is crucial for applications in audio synthesis and sound design, where the modulation of sound envelopes can be used to create desired auditory effects.

Another significant contribution of the paper is the exploration of how carrier frequency affects the perception of roughness. Pressnitzer and McAdams found that higher carrier frequencies tend to reduce the impact of phase changes on roughness perception. This finding indicates a frequency-dependent nature of roughness perception, with lower frequencies being more susceptible to phase-induced variations in roughness. This aspect of their research involved manipulating the carrier frequency of the sound components and observing the changes in perceived roughness. The frequency dependency of roughness perception has important implications for audio engineering and sound design, as it suggests that different frequency bands can be manipulated to achieve specific auditory effects.

The authors also discuss the perceptual mechanisms underlying the observed effects. They propose that the auditory system's sensitivity to phase and envelope shape is likely due to the way these characteristics influence the temporal fine structure and envelope modulation of sounds. Temporal fine structure refers to the rapid fluctuations in the waveform of a sound, while envelope modulation refers to the slower changes in amplitude over time. Both of these aspects are crucial for encoding and decoding auditory information. By altering the phase and envelope shape, the temporal fine structure and envelope modulation are modified, leading to changes in perceived roughness.

Pressnitzer and McAdams' study employs a combination of psychophysical experiments and theoretical analysis to uncover the intricate relationships between phase, envelope shape, and roughness perception. The psychophysical experiments involved presenting participants with carefully controlled sound stimuli and asking them to rate the perceived roughness. Theoretical analysis was used to interpret the results and propose mechanisms underlying the observed effects. This comprehensive approach allows the authors to draw robust conclusions about the factors influencing roughness perception.

In conclusion, the paper "Two-Phase Effects in Roughness Perception" by Daniel Pressnitzer and Stephen McAdams provides substantial insights into the factors that contribute to acoustic roughness. The study's rigorous methodology and comprehensive analysis shed light on the significant roles of phase relationships and envelope shapes in shaping our auditory experiences. These findings have far-reaching implications for fields such as audio signal processing, music production, and auditory research, where understanding and controlling roughness can enhance sound quality and listener experience. This research serves as a foundational work that bridges the gap between theoretical concepts of roughness and practical auditory applications, paving the way for future studies to further explore this intricate auditory phenomenon.

3

Review of "Psychoacoustical Roughness: Implementation of an Optimized Model" by P. Daniel and R. Weber

The paper ”Psychoacoustical Roughness: Implementation of an Optimized Model” by P. Daniel and R. Weber, published in Acustica, vol. 83, 1997, provides an in-depth exploration of the psychoacoustical phenomenon of roughness and presents a highly refined model for its quantification. This work stands out for its meticulous approach to understanding and modeling the perceptual nuances of roughness, an auditory attribute that significantly influences the character and quality of sound.

Daniel and Weber begin by establishing the context of their study within the broader field of psychoacoustics. Roughness is characterized as a perceptual sensation arising from rapid amplitude or frequency modulations in a sound signal. It is a crucial component of timbre, contributing to the perceived harshness or smoothness of sounds. This attribute is especially relevant in diverse applications, from music production to environmental noise assessment. Despite its importance, previous models of roughness have had limitations in accurately capturing the perceptual subtleties of this phenomenon, motivating the authors to develop a more precise and comprehensive model.

The heart of the paper is the development of an optimized model for estimating roughness. The authors critically review existing models, highlighting their strengths and shortcomings. Notably, they point out that many prior models fail to account for the complex interactions between different sound components and the non-linearities in human auditory perception. Daniel and Weber propose an improved model that integrates these factors more effectively.

Their model is grounded in a detailed representation of auditory processing mechanisms, particularly focusing on how the auditory system integrates rapid amplitude fluctuations. Key to their approach is the consideration of modulation frequency and depth, as well as the interaction between different spectral components. The model is structured to reflect the temporal dynamics of sound, recognizing that roughness perception is sensitive to these changes over time.

To validate their model, Daniel and Weber conduct a series of rigorous psychophysical experiments. These experiments involve presenting participants with a range of sound stimuli that systematically vary in modulation frequency and depth. Participants rate the roughness of these sounds, providing empirical data that the authors use to fine-tune their model.

The experimental setup is carefully designed to isolate the effects of different variables on roughness perception. Sounds with different carrier frequencies, modulation indices, and temporal envelopes are tested, ensuring a comprehensive dataset. The authors employ advanced statistical techniques to analyze this data, comparing the model's predictions with actual perceptual responses. This process of empirical validation is crucial for demonstrating the model's reliability and accuracy.

Beyond empirical validation, the paper offers significant theoretical insights into the auditory processing mechanisms underlying roughness perception. Daniel and Weber propose that roughness arises from the temporal integration of rapid amplitude fluctuations by the auditory system. They highlight the role of the cochlea and neural encoding mechanisms in processing these fluctuations. According to their theory, the temporal resolution of auditory neurons plays a critical role in encoding these rapid changes, influencing the perception of roughness.

This neurophysiological perspective provides a deeper understanding of the biological basis of roughness. The authors suggest that roughness perception involves both peripheral and central auditory processes, where initial encoding occurs in the cochlea, and further integration happens at higher auditory centers. This dual-level processing model offers a comprehensive framework for understanding how roughness is perceived and encoded by the auditory system.

Daniel and Weber discuss several practical applications of their optimized roughness model. In audio signal processing, their model can inform the development of algorithms for noise reduction and sound enhancement. By accurately predicting and manipulating roughness, these algorithms can improve the clarity and quality of audio signals. In music production, the model offers a tool for creating sounds with specific roughness characteristics, allowing composers and sound designers to craft more expressive and engaging audio experiences.

The model also holds significant potential in the field of auditory research. It provides a robust framework for studying the perceptual and physiological aspects of roughness, facilitating further investigations into how different auditory phenomena interact and affect perception. Researchers can use this model to explore various aspects of auditory processing, from basic neural mechanisms to complex perceptual phenomena.

"Psychoacoustical Roughness: Implementation of an Optimized Model" by P. Daniel and R. Weber represents a substantial advancement in the field of psychoacoustics. The authors' comprehensive approach to model development, empirical validation, and theoretical analysis sets a new standard for research on roughness perception. Their work not only enhances our understanding of auditory perception but also provides practical tools and insights for various applications in audio technology and auditory research.

By integrating detailed auditory processing mechanisms and validating the model with rigorous psychophysical experiments, Daniel and Weber have created a robust and reliable tool for estimating roughness. This model stands as a foundational reference in psychoacoustics, offering valuable insights and applications that will undoubtedly influence future research and development in auditory science and audio engineering. The paper underscores the importance of considering both spectral and temporal dynamics in modeling auditory perception, paving the way for further advancements in the understanding and manipulation of complex sound attributes.

4

Creation of MATLAB Simulation for Acoustic Roughness Analysis

4.1. Motivation

The motivation behind creating the MATLAB simulation for acoustic roughness analysis stems from the need to gain a deeper understanding of the complex interactions between spectral and temporal sound characteristics that contribute to the perception of roughness. Acoustic roughness, characterized by rapid amplitude or frequency fluctuations that produce a sensation of harshness, is a critical auditory attribute influencing how we perceive sounds. Despite its importance, the intricate factors affecting roughness perception, such as phase relationships and envelope shapes, are not fully understood.

The simulation was designed to address this gap by providing a hands-on, interactive tool that allows researchers and engineers to explore these factors in detail. By enabling real-time manipulation of sound parameters and visualizing their effects, the simulation offers a unique opportunity to study the impact of phase modulation, frequency modulation, and temporal structures on perceived roughness. This is particularly relevant for applications in audio signal processing, music production, and auditory research, where controlling and optimizing sound roughness can enhance overall sound quality and listener experience. The simulation thus serves as both a research instrument and a practical tool for advancing our understanding of acoustic roughness and its applications.

4.2. Initialization

The initial step in the simulation development involves defining the fundamental parameters such as the sampling rate (‘fs‘), duration of the sound signal, and creating a time vector (‘t‘). The sampling rate is set at 44,100 Hz, which is a standard rate for high-quality audio processing, ensuring that the sound is sampled frequently enough to capture all necessary details. The duration is set to 1 second for initial tests, but this can be adjusted dynamically through the simulation interface. The time vector is created using MATLAB’s ‘linspace‘ function, which generates linearly spaced points over the specified interval. This time vector (‘t‘) is essential for generating the sound waveforms and performing subsequent analyses.

4.3. Tone Generation

Once the basic parameters are established, the next step is generating phase amplitude modulation (pAM) tones using the ‘generate_pAM_tone‘ function. This function takes the phase (‘phi‘), time vector (‘t‘), center frequency (‘fc‘), and modulation frequency (‘fm‘) as inputs to create a complex tone. The function generates a waveform composed of three sinusoidal components with different frequencies: ‘fc - fm‘, ‘fc‘, and ‘fc + fm‘. The phase shift is applied to the central component. This pAM tone serves as the primary signal for analyzing roughness, as it incorporates the crucial elements of phase and frequency modulation that affect roughness perception.

4.4. Auditory Filter Bank Simulation

To simulate the peripheral auditory processing, the simulation employs a series of bandpass filters designed to mimic the frequency resolution of the human auditory system. The filter bank parameters include the number of filters (‘numFilters‘), the range of frequencies (‘lowFreq‘ to ‘highFreq‘), and the center frequencies of the filters (‘centerFreqs‘). The filters are designed using the ‘butter‘ function, which creates Butterworth bandpass filters. Each filter is tuned to a specific frequency band, ensuring that the full range of auditory frequencies is covered. The filtered signals are obtained by applying these bandpass filters to the generated pAM tone, simulating how the human ear processes different frequency components.

4.5. Envelope Calculation

After filtering the signal, the next critical step is calculating the envelope of each filtered signal using the Hilbert transform. The envelope represents the amplitude variations of the signal over time, which are crucial for roughness perception. The ‘hilbert‘ function is used to compute the analytic signal, from which the envelope is derived. The absolute value of the Hilbert transform gives the instantaneous amplitude of the signal, which is then used to calculate the root mean square (RMS) value. This RMS value is a measure of the roughness of the signal, as it captures the intensity of the amplitude fluctuations.

4.6. Roughness Estimation

The roughness of each filtered signal is estimated based on the RMS value of its envelope. This estimation is performed using the ’calculate_roughness’ function, which takes the envelope of the signal as input and computes its RMS value. The roughness values of all filtered signals are then summed to obtain the total roughness of the original pAM tone. This comprehensive measure of roughness is crucial for understanding how different sound components and their interactions contribute to the overall perception of roughness.

4.7. Interactive Interface

The MATLAB simulation features an interactive interface that allows users to manipulate the phase, modulation frequency, and duration of the pAM tone in real-time. This interface is created using MATLAB’s GUI development environment. The interface includes sliders for adjusting the phase (‘phaseSlider‘), modulation frequency (‘fmSlider‘), and duration (‘durationSlider‘). Each slider is accompanied by a text label that displays the current value, providing immediate feedback to the user.

The interactive interface also includes a figure window divided into panels for controls and plots. The control panel houses the sliders and their labels, while the plot panel contains axes for displaying the waveform, roughness bar plot, and spectrogram. The ‘guidata‘ function is used to store and manage the UI elements and parameters, ensuring seamless interaction and updates.

4.8. Real-Time Updates

To ensure real-time updates, the simulation employs listeners attached to the sliders. These listeners trigger the ‘updateSimulation‘ function whenever a slider value changes. The ‘updateSimulation‘ function retrieves the current slider values, updates the time vector if the duration has changed, regenerates the pAM tone with the new parameters, and re-applies the filter bank. The function then recalculates the roughness values, updates the plots, and displays the new spectrogram. This dynamic updating mechanism allows users to observe the immediate effects of parameter changes on the waveform, roughness, and spectrogram.

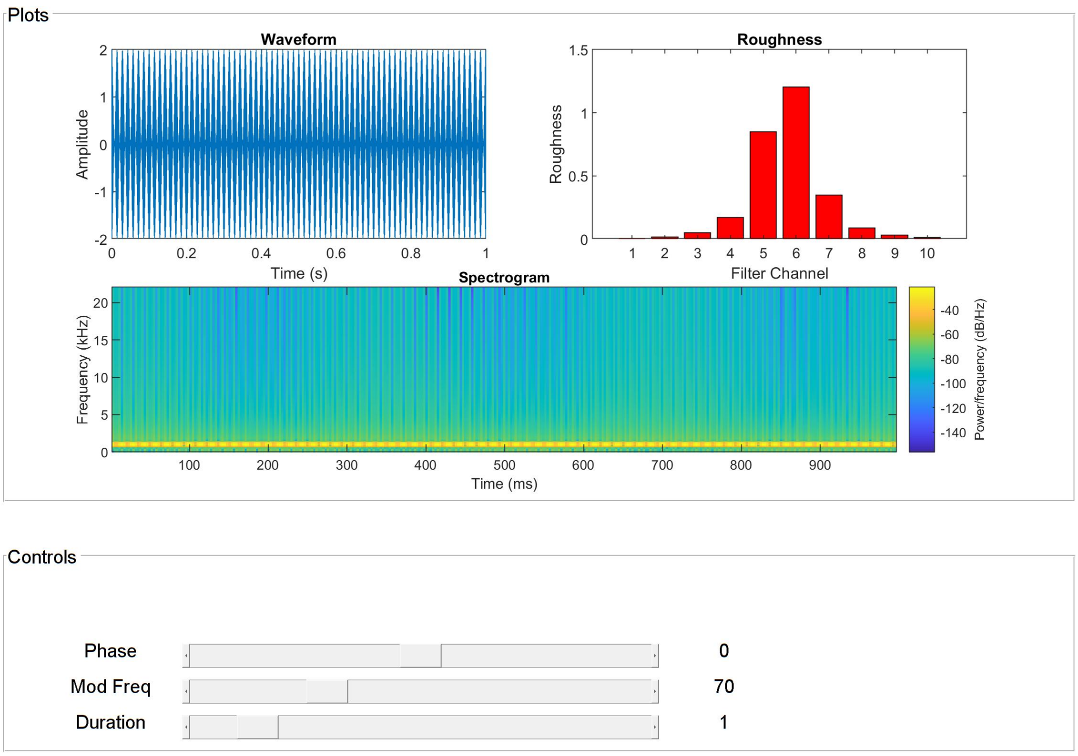
4.9. Applications

The MATLAB simulation for acoustic roughness serves multiple functions. Primarily, it provides a hands-on tool for researchers and engineers to explore the effects of phase, modulation frequency, and duration on roughness perception. By allowing real-time manipulation and visualization of these parameters, the simulation facilitates a deeper understanding of the underlying mechanisms of roughness. The spectrogram plot, in particular, offers a visual representation of how the frequency components and their interactions evolve over time, highlighting the spectral and temporal characteristics that contribute to roughness.

The purpose of this simulation extends beyond academic research. In practical applications, such as audio signal processing, music production, and hearing aid design, understanding and controlling roughness can significantly enhance sound quality and listener experience. For instance, in music production, the ability to manipulate phase and envelope shapes can help create desired auditory textures, making compositions more engaging and dynamic. In hearing aid design, minimizing roughness in processed sounds can improve clarity and comfort for users.

By integrating advanced features and providing an interactive platform, this MATLAB simulation bridges the gap between theoretical research and practical applications. It empowers users to experiment with sound parameters, visualize their effects, and gain valuable insights into the complex phenomenon of acoustic roughness. As such, it stands as a valuable tool for both educational and professional use in the fields of auditory science and audio engineering.

In summary, the creation of the MATLAB simulation involves a series of well-defined steps, each contributing to the overall functionality and effectiveness of the tool. From initializing parameters and generating pAM tones to simulating auditory processing and providing an interactive interface, every aspect of the simulation is designed to enhance our understanding of acoustic roughness. This detailed exploration of the simulation's development and applications underscores its significance in advancing auditory research and improving audio technologies.



5
Conclusion

The capstone project on acoustic roughness has provided a comprehensive exploration of the intricate relationships between spectral and temporal characteristics of sound and their impact on perceived roughness. By delving into the seminal research by Daniel Pressnitzer and Stephen McAdams, we have gained valuable insights into how phase modulation, envelope shape, and frequency content contribute to the complex auditory experience of roughness. The development of an advanced MATLAB simulation was a pivotal component of this project, enabling real-time manipulation and visualization of key sound parameters. This simulation serves as a powerful tool for researchers and engineers, facilitating a deeper understanding of acoustic roughness and its practical applications.

Throughout the project, each step in the simulation's creation was meticulously designed to enhance functionality and user experience. From initializing fundamental parameters and generating phase amplitude modulation tones to simulating auditory processing with a filter bank and providing an interactive interface, every aspect was crafted to provide a robust and flexible platform for analyzing acoustic roughness. The real-time updating mechanism ensures that users can immediately observe the effects of parameter changes, making the simulation a dynamic and engaging tool for both research and education.

The insights gained from this project have significant implications for various fields, including audio signal processing, music production, and hearing aid design. Understanding and controlling roughness can lead to enhanced sound quality, more engaging musical compositions, and improved auditory experiences for hearing aid users. By bridging the gap between theoretical research and practical applications, this project has not only advanced our knowledge of acoustic roughness but also provided a valuable resource for ongoing and future studies in auditory science and audio engineering.

In conclusion, this capstone project has successfully combined rigorous research with innovative simulation development to explore the phenomenon of acoustic roughness. The MATLAB simulation developed as part of this project stands as a testament to the potential for technology to deepen our understanding of complex auditory phenomena and improve real-world applications. As we continue to refine and expand upon this work, the findings and tools generated here will undoubtedly contribute to further advancements in the field of auditory perception and sound processing.

A
MATLAB Simulation Code

```
% CRISTIAN HERNANDEZ
% EE 399 CAPSTONE PROJECT
% NORTHWESTERN UNIVERSITY

% MATLAB Simulation for Acoustic Roughness Analysis

% Parameters
fs = 44100; % Sampling rate
duration = 1; % Duration in seconds
t = linspace(0, duration, fs * duration);
fc = 1000; % Center frequency
fm = 70; % Modulation frequency

% Generate initial pAM tone
phi = 0;
initial_pAM = generate_pAM_tone(phi, t, fc, fm);

% Parameters for the filter bank
numFilters = 10;
lowFreq = 100;
highFreq = 8000;
centerFreqs = logspace(log10(lowFreq), log10(highFreq), numFilters);

% Design bandpass filters
filters = cell(numFilters, 1);
for k = 1:numFilters
    f_low = centerFreqs(k) / sqrt(2);
    f_high = centerFreqs(k) * sqrt(2);
    [b, a] = butter(2, [f_low, f_high] / (fs / 2), 'bandpass');
    filters{k} = {b, a};
end

% Apply the filter bank to the initial pAM tone
filteredSignals = zeros(numFilters, length(initial_pAM));
for k = 1:numFilters
    [b, a] = filters{k}{:};
    filteredSignals(k, :) = filter(b, a, initial_pAM);
end

% Calculate roughness for each filtered signal
roughness_values = zeros(1, numFilters);
for k = 1:numFilters
    roughness_values(k) = calculate_roughness(filteredSignals(k, :));
end

% Total roughness
total_roughness = sum(roughness_values);
fprintf('Estimated Roughness for initial phase %f: %f\n', phi, total_roughness);

```

```matlab
% Play the initial pAM tone
sound(initial_pAM, fs);

% Create a figure window
hFig = figure('Name', 'Acoustic Roughness Simulation', 'NumberTitle', 'off', ...
              'Position', [100, 100, 1000, 700], 'Color', [0.94, 0.94, 0.94]);

% Create panels
controlPanel = uipanel('Title', 'Controls', 'FontSize', 12, ...
                       'BackgroundColor', 'white', 'Position', [0.05, 0.05, 0.9, 0.25]);

plotPanel = uipanel('Title', 'Plots', 'FontSize', 12, ...
                    'BackgroundColor', 'white', 'Position', [0.05, 0.35, 0.9, 0.6]);

% Create sliders for phase, modulation frequency, and duration within the control panel
phaseSlider = uicontrol('Parent', controlPanel, 'Style', 'slider', 'Min', -pi, 'Max', pi, ...
                        'Value', 0, 'Position', [150, 70, 400, 20]);

fmSlider = uicontrol('Parent', controlPanel, 'Style', 'slider', 'Min', 20, 'Max', 200, ...
                     'Value', 70, 'Position', [150, 40, 400, 20]);

durationSlider = uicontrol('Parent', controlPanel, 'Style', 'slider', 'Min', 0.5, 'Max', 5, ...
                           'Value', 1, 'Position', [150, 10, 400, 20]);

% Create text labels for sliders within the control panel
uicontrol('Parent', controlPanel, 'Style', 'text', 'Position', [50, 65, 80, 30], ...
          'String', 'Phase', 'FontSize', 12, 'BackgroundColor', 'white');

uicontrol('Parent', controlPanel, 'Style', 'text', 'Position', [50, 35, 80, 30], ...
          'String', 'Mod Freq', 'FontSize', 12, 'BackgroundColor', 'white');

uicontrol('Parent', controlPanel, 'Style', 'text', 'Position', [50, 5, 80, 30], ...
          'String', 'Duration', 'FontSize', 12, 'BackgroundColor', 'white');

% Create dynamic text to display the current values of the sliders
phaseLabel = uicontrol('Parent', controlPanel, 'Style', 'text', 'Position', [580, 65, 50, 30], ...
                       'String', '0', 'FontSize', 12, 'BackgroundColor', 'white');

fmLabel = uicontrol('Parent', controlPanel, 'Style', 'text', 'Position', [580, 35, 50, 30], ...
                    'String', '70', 'FontSize', 12, 'BackgroundColor', 'white');

durationLabel = uicontrol('Parent', controlPanel, 'Style', 'text', 'Position', [580, 5, 50, 30], ...
                          'String', '1', 'FontSize', 12, 'BackgroundColor', 'white');

% Plot area within the plot panel
waveformAxes = axes('Parent', plotPanel, 'Position', [0.1, 0.55, 0.35, 0.4]);
waveformPlot = plot(waveformAxes, t, initial_pAM);
title(waveformAxes, 'Waveform');
xlabel(waveformAxes, 'Time (s)');
ylabel(waveformAxes, 'Amplitude');

roughnessAxes = axes('Parent', plotPanel, 'Position', [0.55, 0.55, 0.35, 0.4]);
roughnessPlot = bar(roughnessAxes, roughness_values, 'r');
title(roughnessAxes, 'Roughness');
xlabel(roughnessAxes, 'Filter Channel');
ylabel(roughnessAxes, 'Roughness');

spectrogramAxes = axes('Parent', plotPanel, 'Position', [0.1, 0.1, 0.8, 0.35]);
spectrogram(initial_pAM, 256, [], [], fs, 'yaxis');
title(spectrogramAxes, 'Spectrogram');

% Store UI elements and parameters in guidata
handles = struct('phaseSlider', phaseSlider, 'fmSlider', fmSlider, 'durationSlider', durationSlider, ...
                 'waveformPlot', waveformPlot, 'roughnessPlot', roughnessPlot, 'spectrogramAxes', spectrogramAxes, ...
                 't', t, 'fc', fc, 'fs', fs, 'filters', {filters}, 'numFilters', numFilters,
```

```
                    ...
                  'phaseLabel', phaseLabel, 'fmLabel', fmLabel, 'durationLabel', durationLabel);
guidata(hFig, handles);

% Add listeners to the sliders
addlistener(phaseSlider, 'ContinuousValueChange', @(src, evt)updateSimulation(hFig));
addlistener(fmSlider, 'ContinuousValueChange', @(src, evt)updateSimulation(hFig));
addlistener(durationSlider, 'ContinuousValueChange', @(src, evt)updateSimulation(hFig));

% Update function
function updateSimulation(hFig)
    handles = guidata(hFig);

    % Get slider values
    phase = get(handles.phaseSlider, 'Value');
    fm = get(handles.fmSlider, 'Value');
    duration = get(handles.durationSlider, 'Value');

    % Update slider labels
    set(handles.phaseLabel, 'String', num2str(phase, '%.2f'));
    set(handles.fmLabel, 'String', num2str(fm, '%.0f'));
    set(handles.durationLabel, 'String', num2str(duration, '%.2f'));

    % Update time vector based on new duration
    handles.t = linspace(0, duration, handles.fs * duration);

    % Generate pAM tone with new parameters
    new_pAM = generate_pAM_tone(phase, handles.t, handles.fc, fm);

    % Play the new pAM tone
    sound(new_pAM, handles.fs);

    % Update filtered signals
    filteredSignals = zeros(handles.numFilters, length(new_pAM));
    for k = 1:handles.numFilters
        [b, a] = handles.filters{k}{:};
        filteredSignals(k, :) = filter(b, a, new_pAM);
    end

    % Update roughness values
    roughness_values = zeros(1, handles.numFilters);
    for k = 1:handles.numFilters
        roughness_values(k) = calculate_roughness(filteredSignals(k, :));
    end
    total_roughness = sum(roughness_values);

    % Update plots
    set(handles.waveformPlot, 'YData', new_pAM, 'XData', handles.t);
    set(handles.roughnessPlot, 'YData', roughness_values);

    % Plot spectrogram on spectrogramAxes
    cla(handles.spectrogramAxes); % Clear previous spectrogram
    axes(handles.spectrogramAxes); % Set current axes to spectrogramAxes
    spectrogram(new_pAM, 256, [], [], handles.fs, 'yaxis');
    title(handles.spectrogramAxes, 'Spectrogram');

    fprintf('Estimated Roughness for phase %f, mod freq %f, and duration %f: %f\n', phase, fm
        , duration, total_roughness);
end

% Define the pAM tone generation function at the end
function tone = generate_pAM_tone(phi, t, fc, fm)
    tone = 0.5 * cos(2 * pi * (fc - fm) * t) + ...
           cos(2 * pi * fc * t + phi) + ...
           0.5 * cos(2 * pi * (fc + fm) * t);
end

% Calculate the roughness of a signal
function roughness = calculate_roughness(signal)
    roughness = rms(abs(hilbert(signal)));
```

```
end
```