```matlab
% Read the original signal
[signal, fs] = audioread('Pa11.wav');
sound(signal, fs);
pause(length(signal)/fs + 1);
% Generate an echo signal
delay = round(0.1 * fs);   % 0.1 second delay
echo_signal = [zeros(delay, 1); signal(1:end-delay)];   % Create delayed signal
echo_signal = echo_signal * 0.8;   % Echo attenuation
% Add the echo to the original signal to create a noisy signal
noisy_signal = signal + echo_signal;
% Normalize signals to prevent clipping
signal = signal / max(abs(signal));
noisy_signal = noisy_signal / max(abs(noisy_signal));
% Initialize the LMS filter with adjusted parameters
mu = 0.005;                          % Reduced step size
M = 50;                               % Adjusted filter length
lms = dsp.LMSFilter('Length', M, 'StepSize', mu, 'Method', 'Normalized LMS');
% Use the LMS filter to attempt to remove the echo
[output, err] = lms(noisy_signal, signal);   % The desired signal is the original signal
% Normalize output to match the original signal's volume
output = output / max(abs(output)) * max(abs(signal));
% Play the echo-canceled signal
sound(output, fs);
pause(length(output)/fs + 1);
% Plot results to compare
t = (0:length(signal)-1)/fs;   % Time vector for plotting
figure;
subplot(4, 1, 1);
plot(t, signal);
title('Original Signal');
xlabel('Time (s)');
ylabel('Amplitude');


% 读取音频文件
[signal, fs] = audioread('Pa11.wav');

% 计算信号的自相关函数
[Rxx_xcorr, lags] = xcorr(signal, 'biased');

% 绘制自相关函数
figure;
plot(lags/fs, Rxx_xcorr);
title('Autocorrelation of original signal');
xlabel('Time delay (s)');
```

```
ylabel('Autocorrelation amplitude');

% 寻找自相关函数的峰值
[peaks, peak_lags] = findpeaks(Rxx_xcorr, 'MinPeakDistance', 1);

% 排除零延迟峰值
zero_lag_index = find(lags == 0);
if ~isempty(zero_lag_index) && zero_lag_index < length(peaks)
    peaks(zero_lag_index) = [];
    peak_lags(zero_lag_index) = [];
end

% 确定回声的延迟和衰减系数
if isempty(peaks)
    error('No valid peaks found after removing zero-lag peak.');
end

[max_peak, idx] = max(peaks);
echo_peak_lag = peak_lags(idx);  % 回声的延迟，以样本数计
if echo_peak_lag < 1
    error('Echo peak lag must be at least one sample.');
end
echo_delay_seconds = echo_peak_lag / fs;  % 将回声延迟从样本数转换为秒
fprintf('Echo delay is %f seconds\n', echo_delay_seconds);

zero_lag_peak = Rxx_xcorr(find(lags == 0, 1));  % 找到零延迟时的峰值
attenuation_factor = max_peak / zero_lag_peak;
fprintf('Attenuation factor is %f\n', attenuation_factor);

% 设计 FIR 滤波器以消除特定的回声
filter_length = echo_peak_lag + 1;  % 滤波器长度至少要包含回声延迟
b = zeros(filter_length, 1);  % 初始化滤波器系数为 0
b(1) = 1;  % 第一个系数为 1，保留原信号
b(end) = -attenuation_factor;  % 在回声延迟处设置负系数以抵消回声

% 应用 FIR 滤波器来消除回声
filtered_signal = filter(b, 1, signal);

% 播放滤波后的信号
sound(filtered_signal, fs);
```