Digital Guitar Effects in MATLAB®

Signal Processing Implementation & Spectrogram Analysis

Andac Y. Kaya

August 8, 2025

Appendix: Equations and Abbreviations

Abbreviations and Mathematical Symbols	
$x[n]$	Input signal at time step n
$y[n]$	Output signal at time step n
Δt	Sampling period
G	Filter gain or amplification factor
D	Fixed delay (in samples)
$D(t)$	Time-varying delay (modulated by LFO)
$D_{\max}$	Maximum delay time
d	Modulation depth (LFO amplitude factor)
f_{LFO}	Frequency of low-frequency oscillator (Hz)
$a_0,\ a_1$	Attenuation coefficients (feedback/echo gain)
g	Feedback gain in flanger
δ	Semitone shift amount in pitch shifting
α	Time-scaling factor for pitch shifting: $\alpha = 2^{\delta/12}$
ω_0	Instantaneous angular frequency: $\omega_0 = 2\pi f_0(t)$
$f_0(t)$	LFO-modulated center frequency in wah filter
f_{center}	Base center frequency for wah filter
f_{range}	Frequency sweep range of the wah (Hz)
Q	Quality factor (resonance sharpness)

1. Introduction

This project demonstrates a set of audio effects implemented entirely in MATLAB®using signal processing fundamentals. Rather than relying on external plugins or toolboxes, the effects were created through direct manipulation of discrete-time signals, emulating classic guitar effects such as delay, reverb, chorus, flanger, and distortion.

The input is a raw waveform of a clean electric guitar recording. Each effect is applied digitally, and its result is visualized through time-frequency analysis using spectrograms. This report provides a technical summary of each effect along with its audible and visual characteristics. The input guitar signal used in this project is an isolated solo section from an original song by my band, *Buluşalım Rüyalarda- Siyah Orkide.*

This solo was performed, recorded, and processed by the author.

Audio Samples: Audio files for each effect are available here: Google Drive

2. Methodology

Each effect was implemented in MATLAB®using time-domain operations. The following sections outline the core logic behind each effect, accompanied by observations of their impact on the signal's spectrogram.

3. Delay

Effect description: A delayed copy of the signal is added to the original, creating an echo-like result. The delay time is adjustable in samples.

Core equation:

$$y[n] = x[n] + a_0 \cdot x[n - D]$$

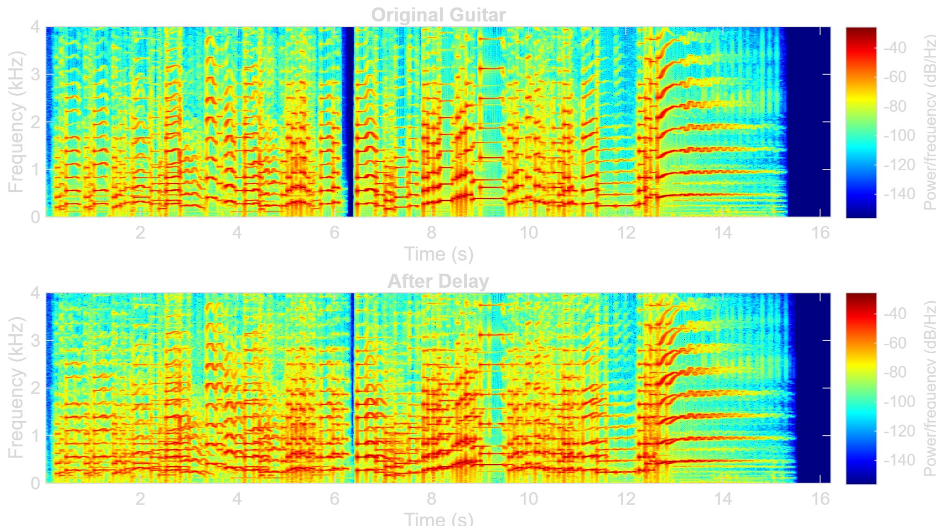

Figure 1: Spectrogram of the guitar signal with delay effect

Spectrogram Observation – Delay

Repeating diagonal traces appear at regular intervals, indicating temporal echoes. Frequency content remains unchanged, but delayed energy bands are visible.

4. Reverb

Effect description: Multiple delayed and attenuated copies of the input are layered to simulate spatial reflections, resulting in a more ambient and sustained sound.

Core equation:

$$y[n] = x[n] + a_1 \cdot (x[n - D] + a_0 \cdot x[n - 2D])$$

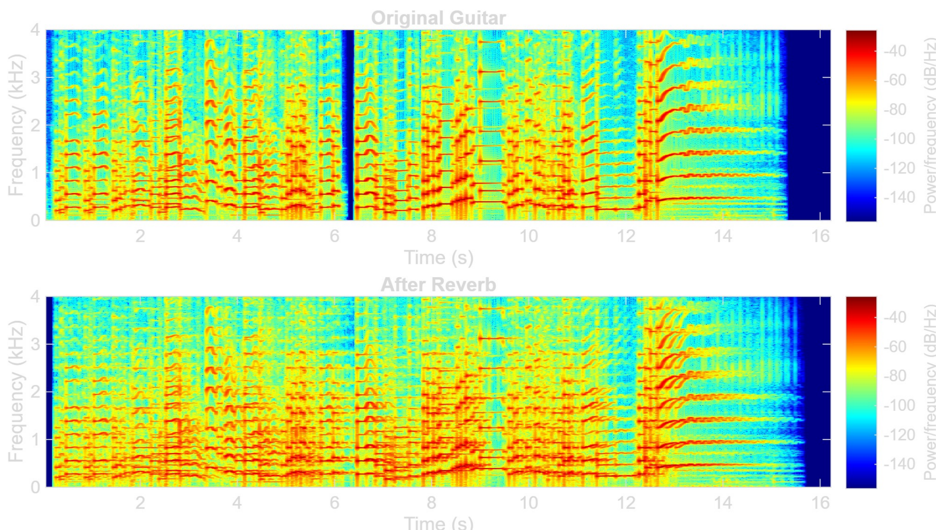

Figure 2: Spectrogram of the guitar signal with reverb effect

Spectrogram Observation – Reverb

Frequency components become smeared over time — vertical edges in the original become blurred. Sustained harmonics and a cloud of energy appear in the higher frequencies.

5. Chorus & Octaver

Effect description: This effect mixes the original signal with pitch-shifted versions, creating a layered, richer tone. Semitone shifts are applied using a time-scaling technique.

Core concept:

$$\alpha = 2^{\delta/12}, \quad y[n] = x[n] + x[\alpha n]$$

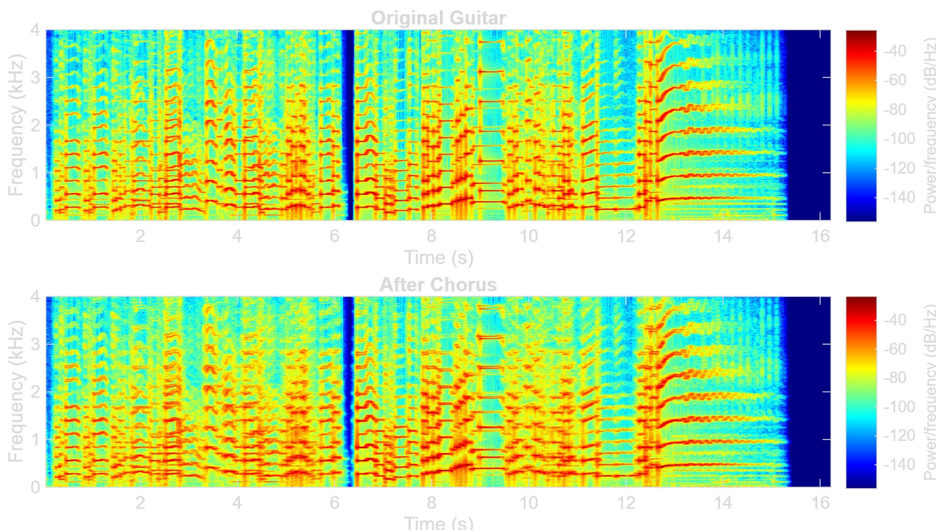

Figure 3: Spectrogram of the guitar signal with chorus effect

Spectrogram Observation – Chorus

Additional frequency bands appear slightly above or below the original harmonics. The result is a thicker, modulated frequency distribution.

6. Flanger

Effect description: A short delay is modulated with a low-frequency oscillator (LFO), creating phase cancellations and comb-filtering effects.

Core equation:

$$y[n] = x[n] + x[n - D(t)] + g \cdot y[n - D(t)]$$

Where:

$$D(t) = D_{\max} \cdot (1 + d \cdot \sin(2\pi f_{\text{LFO}} t))$$

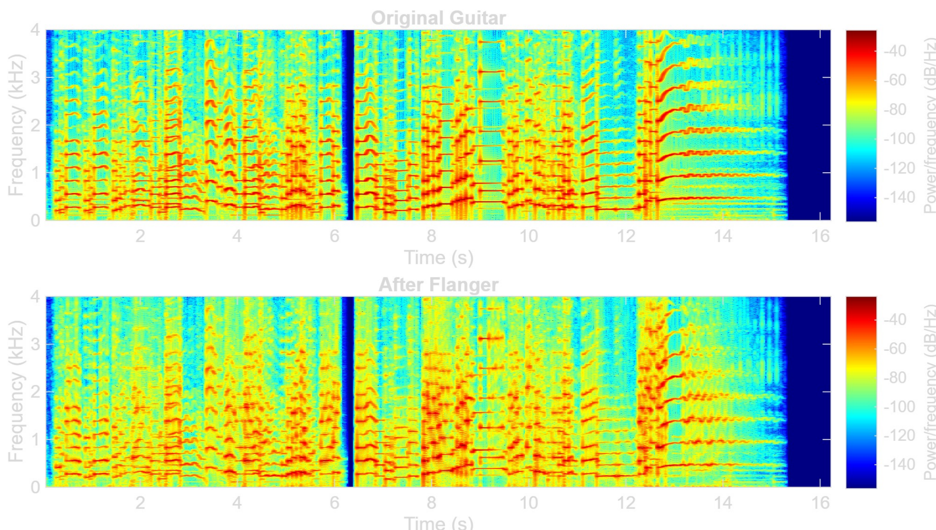

Figure 4: Spectrogram of the guitar signal with flanger effect

Spectrogram Observation – Flanger

Repeating notches and sweeping wave-like interference patterns occur in the frequency domain, caused by LFO-modulated comb filtering.

7. Phaser (Cascaded All-Pass Filters)

Effect description: The phaser effect creates a sweeping, swirling sound by applying a series of all-pass filters to the input signal. These filters shift the phase of different frequency components, and when the original and filtered signals are mixed, phase cancellations occur — creating notches in the frequency spectrum. By modulating the filter coefficients with a low-frequency oscillator (LFO), these notches move over time, producing the characteristic "swooshing" effect of a phaser.

Core equation:

$$y[n] = -a[n] \cdot x[n] + x[n-1] + a[n] \cdot y[n-1]$$

with LFO-driven $a[n] = \text{depth} \cdot \sin(2\pi f_{\text{LFO}} t[n])$. The signal is passed through multiple stages for pronounced effect.

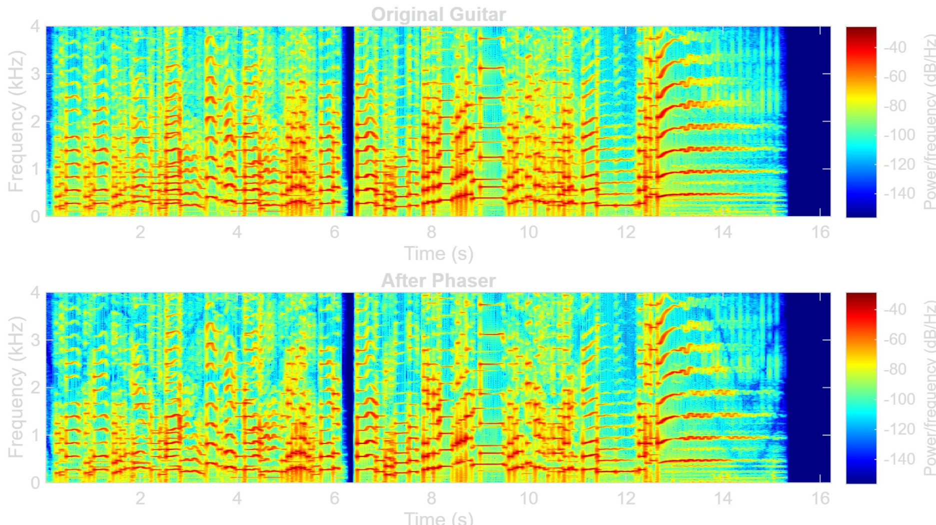

Figure 5: Spectrogram of the guitar signal with phaser effect

Spectrogram Observation – Phaser

The spectrogram shows multiple frequency notches that periodically sweep up and down over time. These moving notches follow a sinusoidal trajectory, driven by the LFO modulation of the all-pass filter stages. The resulting interference pattern produces a distinct phasing texture, with dips in energy appearing and disappearing across the frequency spectrum. This is especially noticeable in sustained notes, where the spectral motion becomes visually and audibly prominent.

8. Distortion

Effect description: Clipping the waveform introduces additional harmonics, resulting in a more aggressive tone.

Core equation (simplified):

$$y[n] = G \cdot x[n] \quad \text{with clipping}$$

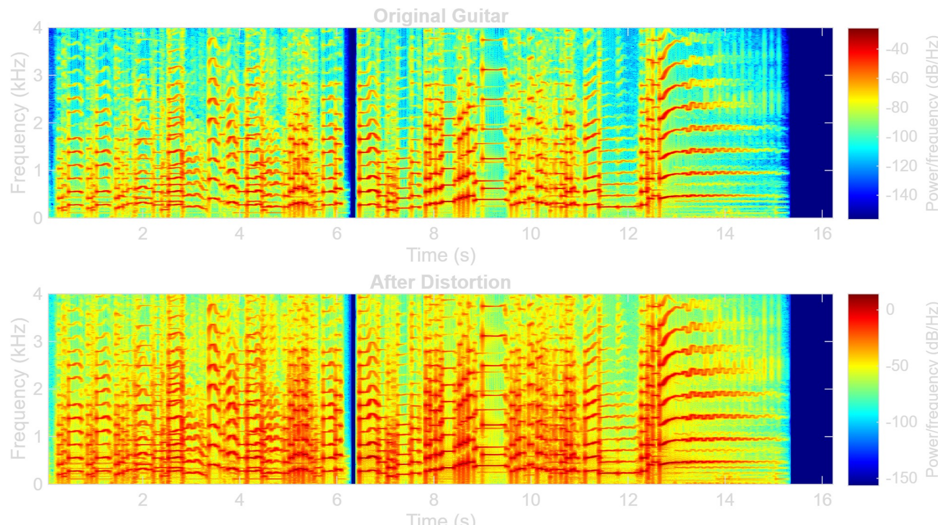

Figure 6: Spectrogram of the guitar signal with distortion effect

Spectrogram Observation – Distortion

High-frequency harmonics become more pronounced. New frequency content appears above the original, resulting in a denser and brighter spectrogram.

9. Low-Pass Filter (2nd order Butterworth using Backward Euler)

Effect description: This effect attenuates high-frequency content while allowing low frequencies to pass through, producing a smoother and warmer sound. A second-order Butterworth filter was implemented digitally using the Backward Euler method, known for its numerical stability in real-time systems.

Core equation: Analog prototype:

$$H(s) = \frac{\omega_c^2}{s^2 + \sqrt{2}\omega_c s + \omega_c^2}$$

where $\omega_c = 2\pi f_c$ is the cutoff angular frequency.

Backward Euler substitution:

$$s \leftarrow \frac{1 - z^{-1}}{\Delta t}$$

Discretized difference equation:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2]$$

Where the coefficients are:

$$\begin{aligned}
a_0 &= 1 + \sqrt{2} G\Delta t + (G\Delta t)^2 \\
a_1 &= -2(1 - (G\Delta t)^2)/a_0 \\
a_2 &= (1 - \sqrt{2} G\Delta t + (G\Delta t)^2)/a_0 \\
b_0 &= (G\Delta t)^2/a_0 \\
b_1 &= 2b_0 \\
b_2 &= b_0
\end{aligned}$$

where $G = 2\pi f_c$ and $\Delta t = \frac{1}{f_s}$ is the sampling period.

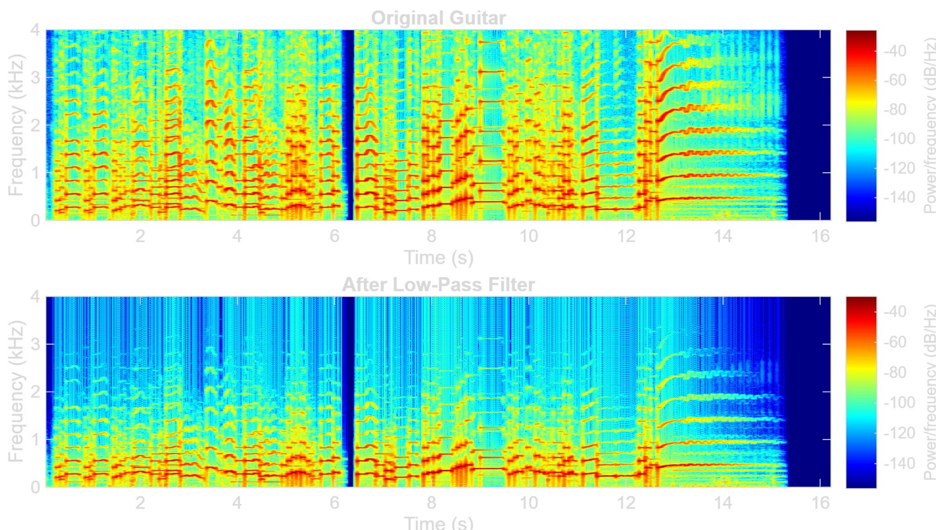

Figure 7: Spectrogram of the guitar signal with low-pass filter

Spectrogram Observation – Low-Pass Filter

High-frequency components are noticeably attenuated, while low-frequency harmonics remain intact or become more prominent. The spectrogram shows reduced energy in upper bands, resulting in a mellowed timbre.

10. High-Pass Filter (2nd order Butterworth using Backward Euler)

Effect description: The high-pass version is derived from the second-order analog Butterworth high-pass filter and converted to digital form using the Backward Euler method for robust numerical stability.

Analog prototype:

$$H(s) = \frac{s^2}{s^2 + \sqrt{2}\omega_c s + \omega_c^2}$$

Backward Euler substitution:

$$s \leftarrow \frac{1 - z^{-1}}{\Delta t}$$

Discretized difference equation:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2]$$

Where the coefficients are:

$$\begin{aligned}
a_0 &= 1 + \sqrt{2} G \Delta t + (G \Delta t)^2 \\
a_1 &= 2((G \Delta t)^2 - 1)/a_0 \\
a_2 &= (1 - \sqrt{2} G \Delta t + (G \Delta t)^2)/a_0 \\
b_0 &= 1/a_0 \\
b_1 &= -2/a_0 \\
b_2 &= 1/a_0
\end{aligned}$$

where again $G = 2\pi f_c$, the analog cutoff angular frequency, and $\Delta t = \frac{1}{f_s}$.

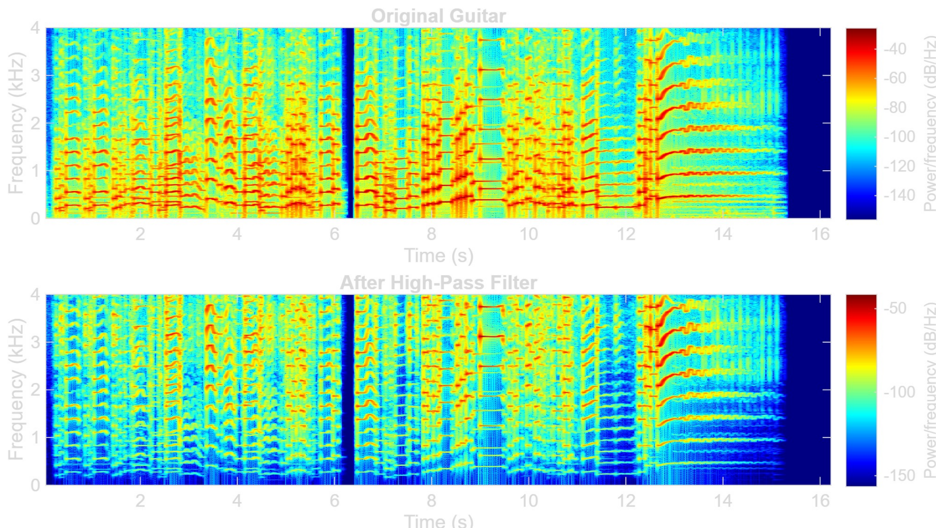

Figure 8: Spectrogram of the guitar signal with high-pass filter

Spectrogram Observation – High-Pass Filter

Low-frequency energy is heavily reduced, shifting the spectral focus to mid and high bands. The resulting sound has more edge and clarity, visible as enhanced high-frequency ridges in the spectrogram.

11. Wah Pedal (2nd-Order Band-Pass Filter using Backward Euler)

Effect description: The Wah effect is achieved by dynamically modulating the center frequency of a band-pass filter. In analog circuits, this is typically done using a potentiometer. Here, a low-frequency oscillator (LFO) modulates the cutoff frequency of a 2nd-order band-pass filter, implemented digitally using the Backward Euler method for time-domain stability.

Analog prototype:

$$H(s) = \frac{\frac{\omega_0}{Q} s}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2}$$

Backward Euler substitution:

$$s \leftarrow \frac{1 - z^{-1}}{\Delta t}$$

Discretized difference equation:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2]$$

Where:

$$f_0(t) = f_{\text{center}} + f_{\text{range}} \cdot \sin(2\pi f_{\text{wah}} \cdot t)$$
$$\omega_0(t) = 2\pi f_0(t)$$

Coefficients:

$$a_0 = 1 + \frac{\omega_0(t)\Delta t}{Q} + (\omega_0(t)\Delta t)^2$$
$$a_1 = \frac{2((\omega_0(t)\Delta t)^2 - 1)}{a_0}$$
$$a_2 = \frac{1 - \frac{\omega_0(t)\Delta t}{Q} + (\omega_0(t)\Delta t)^2}{a_0}$$
$$b_0 = \frac{\omega_0(t)\Delta t/Q}{a_0}, \quad b_1 = 0, \quad b_2 = -b_0$$

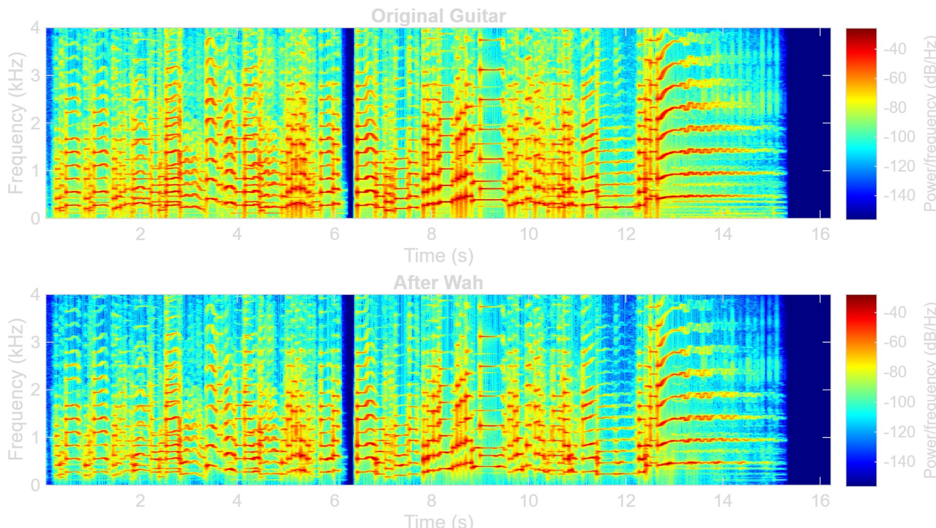

Figure 9: Spectrogram of the guitar signal with wah effect

Spectrogram Observation – Wah Pedal

Sweeping band-pass behavior modulates mid-frequency bands rhythmically. The spectrogram displays alternating boosts and attenuations, characteristic of vocal-like "wah" filtering.

12. Multi-Effect Signal Routing

When using multiple audio effects, the order in which each effect is applied drastically impacts the final tone. Just like in analog pedalboards or amplifier FX chains, digital implementations in MATLAB should follow a signal-aware order that respects the musical nature of each effect.

Effect Chain:

1. **Wah Pedal** – Time-varying filtering based on frequency sweep.
2. **Distortion** – Amplitude clipping for harmonic saturation.
3. **Phaser** – All-pass filtering to create sweeping notches.

4. **Chorus** – Pitch-shifted blend for richness.
5. **Flanger** – Short LFO delay modulation with feedback.
6. **Delay** – Echoing the processed tone.
7. **Reverb** – Ambient reflections added last.

Implementation Logic:

Each processed signal becomes the input to the next effect in the chain:

$$y_{\text{final}}[n] = \text{Reverb}(\text{Delay}(\text{Flanger}(\text{Chorus}(\text{Phaser}(\text{Distortion}(\text{Wah}(x[n])))))))$$

Insight:

This mirrors a guitar routed through a wah pedal and overdrive first (front-end), followed by modulation and ambient effects inserted through an amplifier's FX loop. Such ordering ensures frequency shaping and saturation occur before delay and spatial effects, avoiding smearing or feedback artifacts.

13. Conclusion

All effects were implemented using sample-based processing in MATLAB®. Spectrogram analysis revealed unique and characteristic frequency patterns for each effect. These visual tools not only confirm expected outcomes but also provide insights into how signal processing transforms audio perceptually and spectrally.

Further developments could include real-time implementations or porting these algorithms into VST plugins or embedded DSP platforms.