

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
ELEKTROTEHNIČKI FAKULTET**

**Diplomski sveučilišni studij računarstva**

**VIZUALIZACIJA ZVUKA U STVARNOM VREMENU**

Obrada slike i računalni vid – projektni seminar

Slaven Alerić

Osijek, 2025.

## **SAŽETAK**

U radu je razvijen sustav za vizualizaciju zvuka u stvarnom vremenu koristeći diskretnu Fourierovu transformaciju (DFT) i biblioteku p5.js/p5.sound. Implementirano je pet komplementarnih prikaza: **Bars** (logaritamski spektar), **Line** (linijski spektar na log-f skali), **Spectrogram** (STFT toplinska karta kroz vrijeme), **Scope** (valni oblik u vremenskoj domeni) i **Circular** (valni oblik u polarnim koordinatama). Također implementiran je i estetski prikaz zvuka u obliku kružnog grafa koji prikazuje vremenski signal u polarnim koordinatama. Sustav podržava reprodukciju vlastitih .mp3 datoteka i nekoliko testnih tonova (sinus, akord, dvostruki sinus i šum) za demonstraciju točnosti same vizualizacije. U radu je objašnjena teorijska podloga za izradu projekta (DFT/FFT, STFT, dB skala), detalji implementacije i evaluacija rezultata prikaza na odabranim testnim signalima.

Ključne riječi: Fourierova transformacija, spektar, vizualizacija zvuka, stvarno vrijeme

## SADRŽAJ

<b>UVOD .....</b>	<b>1</b>
<b>2. TEORIJSKA POZADINA.....</b>	<b>2</b>
2.1. DFT i FFT .....	2
2.2. Frekvencijska razlučivost, prozori i „spectral leakage“ .....	3
2.3. Kratkotrajna Fourierova transformacija (STFT) i spektrogram.....	4
2.4. Logaritamska frekvencijska os i dB-skaliranje.....	5
2.5. Vremensko zaglađivanje (EMA) .....	6
<b>3. IMPLEMENTACIJA .....</b>	<b>7</b>
3.1. Inicijalizacija platna, FFT-a i globalnih postavki.....	7
3.2. Učitavanje i odabir pjesama .....	8
3.3. Parametri analize i vremensko zaglađivanje (EMA) .....	9
3.4. Generiranje prikaza .....	9
3.4.1. Logaritamski Bars prikaz s dB mapiranjem.....	9
3.4.2. Linijski spektar na log-f osi .....	10
3.4.3. Spektrogram .....	11
3.4.4. Prikazi Scope i Circular .....	11
<b>4. ISPITIVANJE I EVALUACIJA.....</b>	<b>12</b>
4.1. Kontrolirani testni signali.....	12
4.1.1. Jedan sinus (440 Hz) .....	12
4.1.2. Dualni sinus (440 Hz i 880 Hz).....	13
4.1.3. Bijeli šum.....	14
4.1.4. Akord C-E-G.....	15
4.2. Evaluacija na različitim žanrovima glazbe .....	16
4.2.1. EDM – „Swedish House Mafia ft. John Martin - Don't You Worry Child“ .....	16
4.2.2. Klasična glazba – „Chopin: Nocturne op. 9 br. 2“ .....	17
4.2.3. Rock/metal – „System Of A Down: Chop Suey!“ .....	18
4.3. Usporedba žanrova (EDM, Rock/Metal, Klasična).....	19
<b>5. OGRANIČENJA I MOGUĆNOSTI NADOGRADNJE .....</b>	<b>20</b>
<b>6. ZAKLJUČAK.....</b>	<b>21</b>
<b>LITERATURA.....</b>	<b>22</b>

## UVOD

Ideja vizualizacije zvuka je mapirati akustični signal u grafičke strukture koje ljudsko oko može lagano interpretirati i razumjeti. U sklopu kolegija zadani je zadatak izraditi algoritam koji u stvarnom vremenu prikazuje spektar zadanih zvuka, primjenjuje Fourierovu transformaciju i koristi neku od grafičkih biblioteka za prikaz zvuka. Vizualizator u ovom radu izvodi analizu zvuka u stvarnom vremenu pomoću p5.js/p5.sound, a rezultat prikazuje kao Bars/Line/Spectrogram/Scope/Circular prikaze.

Polazište projekta je činjenica da se frekvencijski sadržaj diskretnog signala može dobiti diskretnom Fourierovom transformacijom (DFT-om). Budući da je izravni DFT prespor za rad u svakom grafičkom *frameu*, zadatak se oslanja na Brzu Fourierovu transformaciju (FFT). Algoritam smanjuje vrijeme izračuna na  $O(N \log N)$ . Zahvaljujući tome spektar zvuka moguće je kontinuirano pratiti i osvježavati u JavaScriptu.

## 2. TEORIJSKA POZADINA

Cilj ovog vizualizatora zvuka je prikazati raspodjelu energije zvuka po frekvencijama i kako se ona mijenja kroz vrijeme. To uključuje korištenje diskretnе Fourierove analize, kratkotrajne Fourierove transformacije (STFT) za vremensku lokalizaciju, te nekoliko praktičnih koraka za lakše vizualno prepoznavanje (logaritamska frekvencijska os, decibel-skala, zaglađivanje). U ovom dijelu objašnjava se teorija koja stoji iza izrađenog rješenja zadatka i povezanost s parametrima koji se u projektu zaista koriste.

### 2.1. DFT i FFT

Cilj vizualizatora je iz vremenskog signala zvuka izdvojiti informacije po frekvencijama. Diskretna Fourierova transformacija omogućuje upravo to: signal  $x[n]$  duljine  $N$  predstavlja se kao zbroj harmonika različitih frekvencija. Za diskrete, konačne signale DFT je „**potpuna Fourierova reprezentacija signala**“ [2]. Za niz uzoraka zvuka  $x[n]$  duljine  $N$  (uzorkovano frekvencijom  $f_s$ ) diskretna Fourierova transformacija (DFT) definira se kao:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi k / N}, \quad k = 0, \dots, N - 1$$

Inverzna transformacija vraća signal iz spektra:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{+j2\pi kn/N}$$

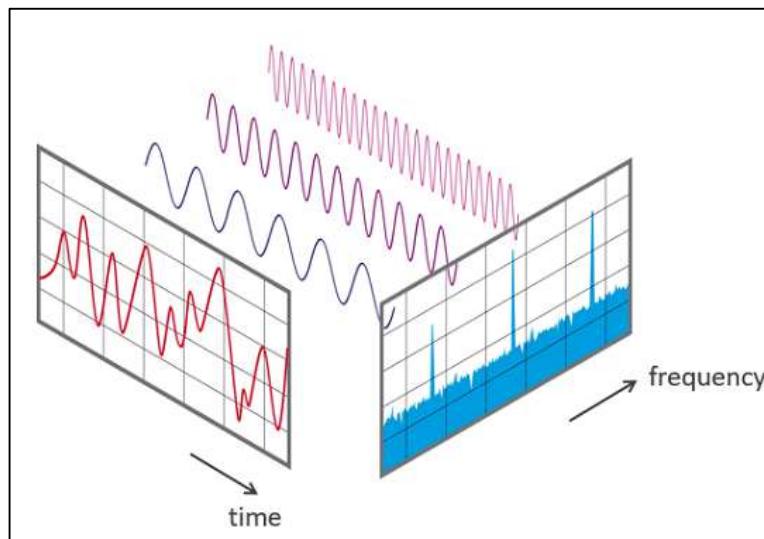
Svaki indeks  $k$  pripada diskretnoj frekvenciji:

$$f_k = \frac{k}{N} f_s, \quad 0 \leq f_k \leq \frac{f_s}{2}$$

U praksi, umjesto izravnog računanja DFT-a (složenost  $O(N^2)$ ), koristi se **brza Fourierova transformacija (FFT)**. Povijesno, najvažnija je Cooley-Tukeyjeva strategija rastavljanja

problema na manje DFT-ove, pri čemu se broj operacija smanjuje (složenost  $O(N \log N)$ ) [3]. FFT omogućuje matematički jednostavno osvježavanje spektra u svakom grafičkom okviru (*frameu*), što je nužno za zadani zadatak kako bi vizualizacija mogla pratiti zvukove i „disati“ u stvarnom vremenu s danom glazbom, odnosno zvukom.

Na slici 1 je prikaz signala u vremenskoj i frekvencijskoj domeni. Lijevo se može vidjeti valni oblik u vremenu (crvena krivulja). On se može rastaviti na zbroj **sinusoidnih komponenti** različitih frekvencija i faza (ljubičaste/plave linije). FFT računa tu razgradnju i na desnoj ploči prikazuje **spektar**: na x osi je **frekvencija**, na y osi **amplituda**. Uski vrhovi odgovaraju pojedinim tonovima/komponentama, a plava podloga predstavlja preostalu energiju ili šum.



Slika 1 Prikaz signala u vremenskoj i frekvencijskoj domeni Izvor: <https://www.nti-audio.com/portals/0/pic/news/FFT-Time-Frequency-View-540.png>

## 2.2. Frekvencijska razlučivost, prozori i „spectral leakage“

Kod diskretnе Fourierove analize promatramo konačan isječak signala. Ako u tom isječku postoji sinus čija frekvencija ne pada točno na frekvencijsku „mrežu“ DFT-a, na rubovima prozora nastaje diskontinuitet. Da bi objasnio taj rubni skok, DFT uzima doprinos iz cijelog frekvencijskog raspona, te umjesto da se dio energije zadržava u jednom binu on se rasprši na više susjednih. Ova pojava se naziva **razmazivanje spektra** (spectral leakage) [4]. Važna posljedica ovoga je da se uz uske vrhove vide i povišene vrijednosti i u susjednim binovima, u primjeru zadatka to će se vidjeti u *Bars* prikazu gdje se za čisti ton često podigne jedan ili dva susjedna stupca, a u *Line* prikazu vrh postane vidljivo širi. Potpuno uklanjanje razmazivanja

**nije moguće** jer energija svakog frekvencijskog sastojka „mora negdje završiti“ u diskretnoj rešetci binova [4]. Razmazivanje se može usmjeravati i ublažavati primjenom **windowinga**.

Razmazivanje se može djelomično smanjiti tako da mjereni isječak prije FFT-a pomnožimo prozorskom funkcijom  $w[n]$  koja na rubovima teži nuli:

$$x_w[n] = x[n] * w[n], \quad n = 0, \dots, N - 1$$

Time rubni diskontinuitet postaje blaži pa se energija manje „razljeva“ u bočne binove [4]. U praksi to znači da se u *Bars* i *Line* prikazu vrh čistog tona zadrži izraženiji u središnjem binu, a susjedni binovi budu niži.

### 2.3. Kratkotrajna Fourierova transformacija (STFT) i spektrogram

Za prikaz promjena u spektru tijekom vremena koristi se proširenje klasične Fourierove transformacije koje se zove **kratkotrajna Fourierova transformacija (STFT)**. Dok standardna DFT pruža globalnu frekvencijsku analizu signala, STFT omogućuje lokalnu analizu frekvencijskog sadržaja u određenim vremenskim isječcima. STFT se izvodi tako da se signal podijeli na segmente, te se na svaki segment primjeni DFT. Tako dobivamo funkciju vremena i frekvencije:

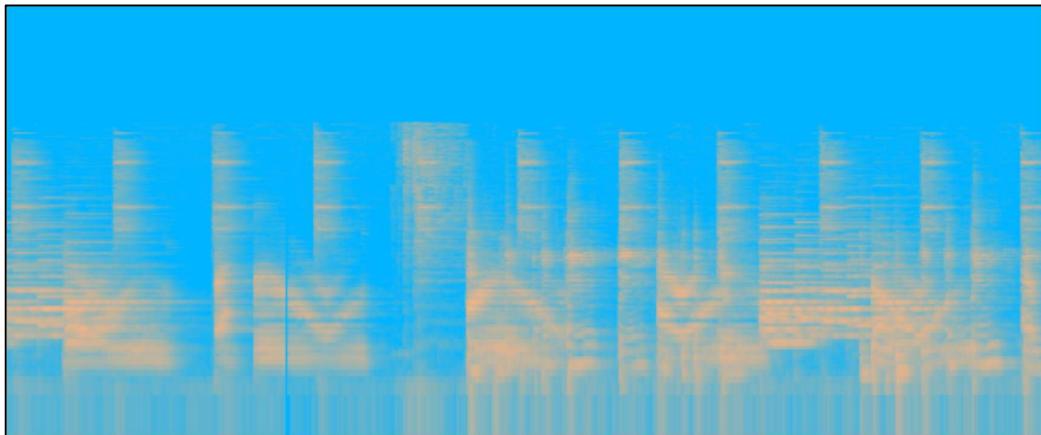
$$STFT\{x[n]\}(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega}$$

gdje je  $w[n]$  prozorska funkcija koja ograničava segment signala na određenom prozoru [5].

Rezultat STFT analize se može prikazati kao **spektrogram** koji je dvodimenzionalna slika u kojoj x-os predstavlja vrijeme, y-os frekvenciju, a boja ili intenzitet amplitudu ili snagu signala na toj frekvenciji i vremenu.

U ovom projektu spektrogram se implementira pomoću *off-screen* platna (p5.Graphics) koji se pri svakom *frameu* ažurira novim stupcem boja, dok se prethodni sadržaj pomiče ulijevo. Boja svakog piksela kodira snagu signala u odgovarajućem frekvencijskom pojasu kao što se vidi na slici 2.

STFT predstavlja kompromis između vremenske i frekvencijske razlučivosti: kraći prozori omogućuju bolju vremensku, a dulji bolju frekvencijsku razlučivost [6].



Slika 2 Prikaz spektrograma tijekom sviranja glazbe

## 2.4. Logaritamska frekvencijska os i dB-skaliranje

Ljudsko uho ne registrira frekvencije na linearan način, već približno **logaritamski** – razlika između 100 Hz i 200 Hz zvuči jednako velika kao između 1000 Hz i 2000 Hz. Zbog toga se logaritamska frekvencijska os često koristi u audio vizualizacijama jer bolje odražava način na koji ljudi doživljavaju razlike u frekvencijama [7].

U ovom projektu korištena je log-frekvencijska skala u svim prikazima frekvencijskog spektra (*Bars, Line, Spectrogram*), što omogućuje da se niske frekvencije detaljnije prikažu, dok se više frekvencije grupiraju u šire pojaseve.

Osim toga, amplituda frekvencijskih komponenti prikazana je u **decibelima (dB)**, koji također slijede logaritamsku mjeru. Decibelska skala definira se kao:

$$A_{dB} = 20 * \log_{10} \left( \frac{A}{A_{ref}} \right)$$

gdje je  $A$  amplituda trenutnog signala, a  $A_{ref}$  referentna amplituda [8].

Korištenjem log-skale po frekvenciji i logaritamske amplitude (dB), spektar postaje vizualno razumljiviji i korisniji za audio interpretacije. U ovom projektu se sirove vrijednosti iz FFT mapiraju u raspon dB skale (-60dB do 0dB). Time se naglašavaju prisutne komponente dok se manje komponente poput šuma potiskuju.

## 2.5. Vremensko zaglađivanje (EMA)

Prilikom vizualizaciji zvuka u stvarnom vremenu često se javlja nepoželjno treperenje ili prevelika osjetljivost na trenutne skokove u amplitudi što u nekim slučajevima može značajno otežati interpretaciju. Da bi se vizualizacija stabilizirala i bila jednostavnija za praćenje koristi se **jednostavni IIR niskopropusni filter prvog reda** (tzv. **eksponencijalno pomično srednjavanje, EMA**), zadan rekurzijom  $y[n] = \alpha x[n] + (1 - \alpha)y[n - 1]$ ,  $0 < \alpha \leq 1$  [9]. Za svaki novi uzorak spektra ili vremenskog signala primjenjuje se formula:

$$y_t = \alpha * x_t + (1 - \alpha) * y_{t-1}$$

gdje je:  $x_t$  trenutna vrijednost,  $y_t$  izglađena vrijednost,  $\alpha$  koeficijent prilagodbe.

U projektu korisnik ima mogućnost sam podešiti razinu zaglađivanja (*smoothing*) putem **slidera** na vrhu web stranice, što omogućuje prilagodbu između brzog odziva i stabilnosti vizualizacije.

Vrijednost  $\alpha$  se onda računa kao  $1 - \text{smoothing}$  gdje je *smoothing* između 0 i 0,95.

### 3. IMPLEMENTACIJA

Implementacija rješenja zadatka vizualizacije zvuka u stvarnom vremenu se temelji na sustavu koji je razvijen u **JavaScriptu** koristeći **p5.js** i njezinu zvučnu ekstenziju **p5.sound** koje omogućuju jednostavno povezivanje zvuka i grafike u web okruženju. Projekt je modularno strukturiran unutar jedne JS i jedne HTML datoteke te sadrži logički odvojene cjeline. U nastavku je analiza strukture koda i objašnjenje ključnih funkcija i algoritama korištenih u projektu.

#### 3.1. Inicijalizacija platna, FFT-a i globalnih postavki

Aplikacija na početku definira sve parametre i stanja kao što se vidi na slici 3: rezoluciju FFT-a, broj bars stupaca, raspon dB skale, dimenzije off-screen platna za spektrogram i varijable koje se pamte kroz *frameove*. Funkcije *setup()* i *rebuildFFT()* su jezgra inicijalizacije. U *setup()* se kreira *canvas* i pomoćno grafičko platno *specG* za spektrogram. U *rebuildFFT()* se sa parametrima stvara p5.FFT i resetira EMA memorija kako bi promjene „*smoothinga*“ odmah stupile na snagu.

```
const CFG = {
  bins: 1024, barsCount: 48, dbMin: -60, dbMax: 0, specW: 640, specH: 256, axisMargin: 16
};

let song = null; let fft = null; let view = 'bars'; let specG; let currentName = null; let smoothUI = 0.85;
let prevSpec = null; let prevWave = null;

const $ = (sel) => document.querySelector(sel);

function setup() {
  const wrap = document.querySelector('.canvasWrap') || document.body;
  const c = createCanvas(window.innerWidth, window.innerHeight - 56);
  c.parent(wrap);
  angleMode(RADIANS);
  strokeJoin(ROUND); strokeCap(ROUND);
  noFill();

  rebuildFFT();

  specG = createGraphics(CFG.specW, CFG.specH);
  specG.colorMode(HSB, 255);
}

function rebuildFFT() {
  fft = new p5.FFT(smoothUI, CFG.bins);
  if (song) fft.setInput(song);

  prevSpec = null;
  prevWave = null;
}
```

Slika 3 Definiranje stanja, funkcije *setup()* i *rebuildFFT()*

*draw()* je funkcija koja jednostavno briše pozadinu i ovisno o *view* poziva odgovarajući prikaz. Ukoliko zvuk nije učitan daje se uputa korisniku koje opcije ima na izbor i što može napraviti.

```
function draw() {
    background(0);

    if (!song || !fft) {
        drawCenteredHint('Odaberite pjesmu iznad (ili dodaj .mp3) tipka V mijenja prikaz , Space = Play/Pause');
        return;
    }

    switch (view) {
        case 'bars':      drawSpectrumBars();      break;
        case 'line':       drawSpectrumLine();       break;
        case 'spectrogram': drawSpectrogram();       break;
        case 'scope':      drawScope();             break;
        case 'circular':   drawCircularWaveform();  break;
    }
}
```

Slika 4 Funkcija *draw()*

### 3.2. Učitavanje i odabir pjesama

Učitavanje pjesama riješeno je na dva načina koja će biti prikazana na slici 5, odnosno u sljedećim linijama koda, a to su (1) automatsko listanje .mp3 datoteka iz *songs/* direktorija, a (2) ručni upload preko *<input type = „file“>*. Kod je organiziran tako da UI ostane jednostavan. Za svaku pjesmu generira se gumb bilo ona bila pronađena u direktoriju ili dodana od strane korisnika. Svaki gumb onda poziva istu funkciju *pickSong()* koja učita zvuk pomoću funkcije *loadSound*, obnovi FFT i pokrene reprodukciju. Cijeli tok od klika na gumb za zvuk i vizualizacije je direktni i bez ikakvih duplicitiranih stanja.

```
async function buildButtonsFromDir() {
    const box = $('#songButtons');
    if (!box) return;
    box.innerHTML = '';
    try {
        const res = await fetch('songs/');
        if (!res.ok) throw new Error('no listing');
        const html = await res.text();
        const doc = new DOMParser().parseFromString(html, 'text/html');
        const links = [...doc.querySelectorAll('a')]
            .map(a => a.getAttribute('href'))
            .filter(h => h && h.toLowerCase().endsWith('.mp3'))
            .map(h => decodeURIComponent(h.replace(/\^.\?\\//, '')));

        if (!links.length) {
            box.innerHTML = '<em style="color:#9aa0aa">Nema .mp3 u /songs. Dodaj datoteke ili koristi + Dodaj .mp3</em>';
            return;
        }
        links.forEach(file => {
            const full = `songs/${file}`;
            const btn = document.createElement('button');
            btn.className = 'btn';
            btn.textContent = file;
            btn.onclick = () => pickSong(full, file, btn);
            box.appendChild(btn);
        });
    } catch {
        box.innerHTML = '<em style="color:#9aa0aa">Directory listing nije dostupan + koristi + Dodaj .mp3</em>';
    }
}
```

Slika 5 Funkcija *buildButtonsFromDir()*

### 3.3. Parametri analize i vremensko zaglađivanje (EMA)

Vizualna stabilnost postignuta je kombinacijom ugrađenog zaglađivanja u p5.FFT, koji je i prvi argument konstruktora, i EMA filtracije preko uzastopnih *frameova*. Slider za *Smoothing* mijenja *smoothUI* u rasponu 0 – 0,95 i rekonstruira fft. Pojednostavljenje EMA-a je izračunati novu vrijednost koristeći jednadžbu koja je navedena u teorijskom opisu. U kodu je to napravljeno u funkciji *smoothArray()* (slika 6) koja odvojeno pamti memoriju za spektar i za vremenski signal.

```
function smoothArray(arr, kind) {
  const alpha = 1 - smoothUI;
  if (kind === 'spec') {
    if (!prevSpec || prevSpec.length !== arr.length) prevSpec = new Array(arr.length).fill(0);
    const out = new Array(arr.length);
    for (let i = 0; i < arr.length; i++) out[i] = alpha * arr[i] + (1 - alpha) * prevSpec[i];
    prevSpec = out;
    return out;
  } else {
    if (!prevWave || prevWave.length !== arr.length) prevWave = new Array(arr.length).fill(0);
    const out = new Array(arr.length);
    for (let i = 0; i < arr.length; i++) out[i] = alpha * arr[i] + (1 - alpha) * prevWave[i];
    prevWave = out;
    return out;
  }
}
```

Slika 6 Funkcija *smoothArray(...)*

### 3.4. Generiranje prikaza

#### 3.4.1. Logaritamski Bars prikaz s dB mapiranjem

*Bars* prikaz najprije dohvata spektar (*fft.analyze(CFG.bins)*), zatim ga zagladi EMA-om i grupira u logaritamske frekvencijske pojaseve. Grupiranje je provedeno tako da se za svaki FFT bin izračuna pripadna fizička frekvencija  $f_i$  i logaritamski normira na interval [0,1]. Zatim se preslikava u indeks pojasa. Svaki pojaz se računa ( $* i$ ) i naknadno dijeli s ukupnom težinom. Na kraju vrijednost (0 – 255) se mapira u odabrani dB raspon i vizualizira kao stupac. Kod funkcije se nalazi na slici 7.

```

function drawSpectrumBars() {
  const raw = fft.analyze(CFG.bins);
  const spec = smoothArray(raw, 'spec');

  const bands = CFG.barsCount;
  const nyq = sampleRate() / 2;

  const bins = new Array(bands).fill(0);
  const wts = new Array(bands).fill(0);

  for (let i = 1; i < spec.length; i++) {
    const f = i * nyq / (spec.length - 1);
    const t = Math.log10(Math.max(1, f)) / Math.log10(nyq);
    let b = Math.floor(t * bands);
    b = constrain(b, 0, bands - 1);
    bins[b] += spec[i] * i;
    wts[b] += i;
  }

  const margin = CFG.axisMargin;
  const W = width - margin*2;
  const H = height - margin*3 - 60;
  const barW = W / bands;

  noStroke();
  for (let b = 0; b < bands; b++) {
    const v = wts[b] ? (bins[b]/wts[b]) : 0;
    const db = map(v, 0, 255, CFG.dbMin, CFG.dbMax);
    const h = constrain((db - CFG.dbMin) / (CFG.dbMax - CFG.dbMin), 0, 1);
    const x = margin + b * barW;
    const y = height - margin - h*H;
    fill(149, 202, 255);
    rect(x, y, barW * 0.9, h*H, 5);
  }
  drawFreqRuler(20, 20000, margin, height - margin, W);
}

```

Slika 7 Funkcija *drawSpectrumBars()*

### 3.4.2. Linijski spektar na log-fosi

Prikaz za linijski spektar koristi istu ulaznu logiku (FFT + EMA) kao što se vidi na slici 8, ali svaku točku crta na logaritamski skaliranu x-os pomoću pomoćne funkcije *mapLog*.

```

function drawSpectrumLine() {
  const raw = fft.analyze(CFG.bins);
  const s = smoothArray(raw, 'spec');

  const margin = CFG.axisMargin;
  const y0 = height - margin, y1 = 100 + margin;

  noFill(); stroke(180, 220, 200); strokeWeight(2);
  beginShape();
  for (let i = 1; i < s.length; i++) {
    const f = indexToFreq(i, s.length);
    const x = mapLog(f, 20, 20000, margin, width - margin);
    const db = map(s[i], 0, 255, CFG.dbMin, CFG.dbMax);
    const v = constrain((db - CFG.dbMin) / (CFG.dbMax - CFG.dbMin), 0, 1);
    const y = y0 - v * (y0 - y1);
    vertex(x, y);
  }
  endShape();
  drawFreqRuler(20, 20000, margin, y0, width - 2*margin);
}

function mapLog(f, fmin, fmax, xmin, xmax) {
  const lf = Math.log10(Math.max(fmin, Math.min(fmax, f)));
  const a = (lf - Math.log10(fmin)) / (Math.log10(fmax) - Math.log10(fmin));
  return xmin + a * (xmax - xmin);
}

```

Slika 8 Funkcija *drawSpectrumLine()* i pomoćna funkcija *mapLog(...)*

### 3.4.3. Spektrogram

Spektrogram se generira na pomoćnom platnu *specG*. Svakim *frameom* cijela slika se pomiče za jedan piksel ulijevo, a desni rub naslika novi stupac boja. Indeksiranje po visini preslikava se logaritamski na indekse FFT-a kako bi nizak raspon frekvencija imao veću vizualnu razlučivost. Boja (HSB) kodira jačinu (0 – 255). Ovaj pristup je računski poprilično jednostavan i „jeftin“, kao što se vidi na slici 9, te je vizualno jasan i lijepo teče kad se gleda uživo.

```
function drawSpectrogram() {
  const s = fft.analyze(512);
  specG.copy(specG, 1, 0, specG.width - 1, specG.height, 0, 0, specG.width - 1, specG.height);

  specG.loadPixels();
  for (let y = 0; y < specG.height; y++) {
    const t = 1 - y/specG.height;
    const idx = Math.floor(Math.pow(t, 3.0) * (s.length - 1));
    const val = s[idx]; // 0..255
    const col = color(val, 180, 255 - val/2);
    specG.set(specG.width - 1, y, col);
  }
  specG.updatePixels();

  const margin = CFG.axisMargin;
  image(specG, margin, 100, width - 2*margin, height - 160);
  drawFreqRuler(20, 20000, margin, height - margin, width - 2*margin);
}
```

Slika 9 Funkcija *drawSpectrogram()*

### 3.4.4. Prikazi Scope i Circular

Za vremensku domenu dovoljno je dohvatiti *fft.waveform()*, provući ga kroz uglađivanje EMA-om i ucertati Scope prikaz koji pokazuje standardni osciloskopski prikaz, dok Circular isti signal razvija u polarne koordinate. Debljina linije u Circular prikazu prilagođava se energiji basa (*getEnergy(20,200)*) čime je dobiven pulsirajući efekt koji dodatno obogaćuje vizualizaciju zvuka u skladu s glazbom

```
function drawScope() {
  const raw = fft.waveform();
  const wave = smoothArray(raw, 'wave');

  const margin = CFG.axisMargin;
  const x0 = margin, x1 = width - margin;
  const y0 = 100 + margin, y1 = height - margin;

  stroke(120); strokeWeight(1); noFill();
  rect(x0, y0, x1-x0, y1-y0);

  stroke(170, 160, 255); strokeWeight(2);
  noFill(); beginShape();
  for (let i = 0; i < wave.length; i++) {
    const x = map(i, 0, wave.length-1, x0, x1);
    const y = map(wave[i], 1, -1, y0, y1);
    vertex(x, y);
  }
  endShape();
}

function drawCircularWaveform() {
  const raw = fft.waveform();
  const wave = smoothArray(raw, 'wave');

  const bass = fft.getEnergy(20, 200) / 255;
  const cx = width/2, cy = height/2 + 20;
  const R = 0.28 * Math.min(width, height);

  stroke(255); strokeWeight( lerp(2, 10, bass) );
  noFill(); beginShape();
  for (let i = 0; i < wave.length; i++) {
    const a = (i / (wave.length-1)) * TWO_PI;
    const r = R + wave[i] * R * 0.45;
    vertex(cx + r * Math.cos(a), cy + r * Math.sin(a));
  }
  endShape(CLOSE);
}
```

Slika 10 Funkcije *drawScope()* i *drawCircularWaveform()*

## 4. ISPITIVANJE I EVALUACIJA

Cilj ispitivanja koje će biti provedeno na izrađenom projektu vizualizacije je provjeriti ponašanje vizualizatora na određenim **poznatim i kontroliranim testnim signalima** i na **stvarnoj glazbi** iz različitih žanrova. Time će se evaluirati točnost prikaza frekvencijskih komponenti, stabilnost vizualizacije (uz EMA) i informativnost svih prikaza.

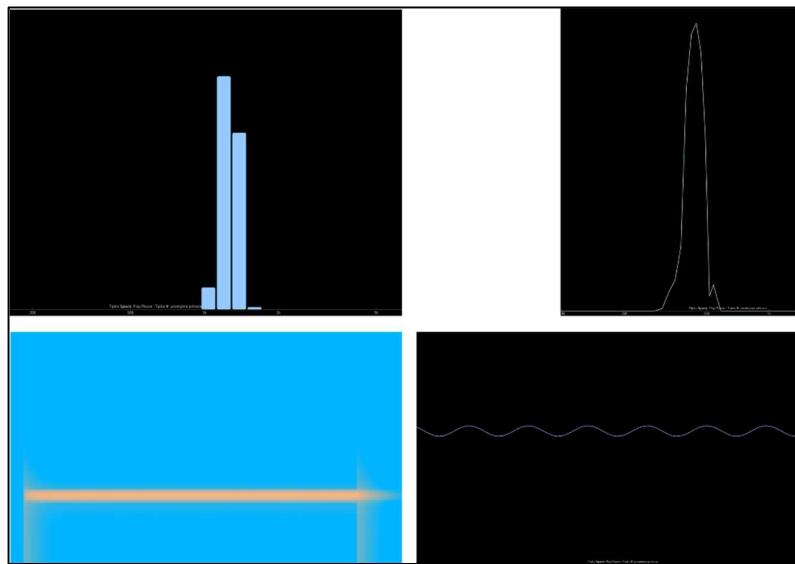
### 4.1. Kontrolirani testni signali

#### 4.1.1. Jedan sinus (440 Hz)

U tablici 1 bit će prikaz testiranja i evaluacije za testni signal jednog sinusa (440 Hz), a dobivene vizualizacije na slici 11.

Prikaz	Očekivanje	Uočeno
<i>Bars</i>	1 stupac oko 440 Hz	1 stupac oko 440 Hz <input checked="" type="checkbox"/>
<i>Line</i>	Uski vrh $\approx$ 440 Hz	Uski vrh $\approx$ 440 Hz <input checked="" type="checkbox"/>
<i>Spectrogram</i>	Tanka vodoravna linija	Tanka vodoravna linija <input checked="" type="checkbox"/>
<i>Scope</i>	Sinusoida	Sinusoida <input checked="" type="checkbox"/>

Tablica 1 Rezultati testiranja – sinus (440 Hz)



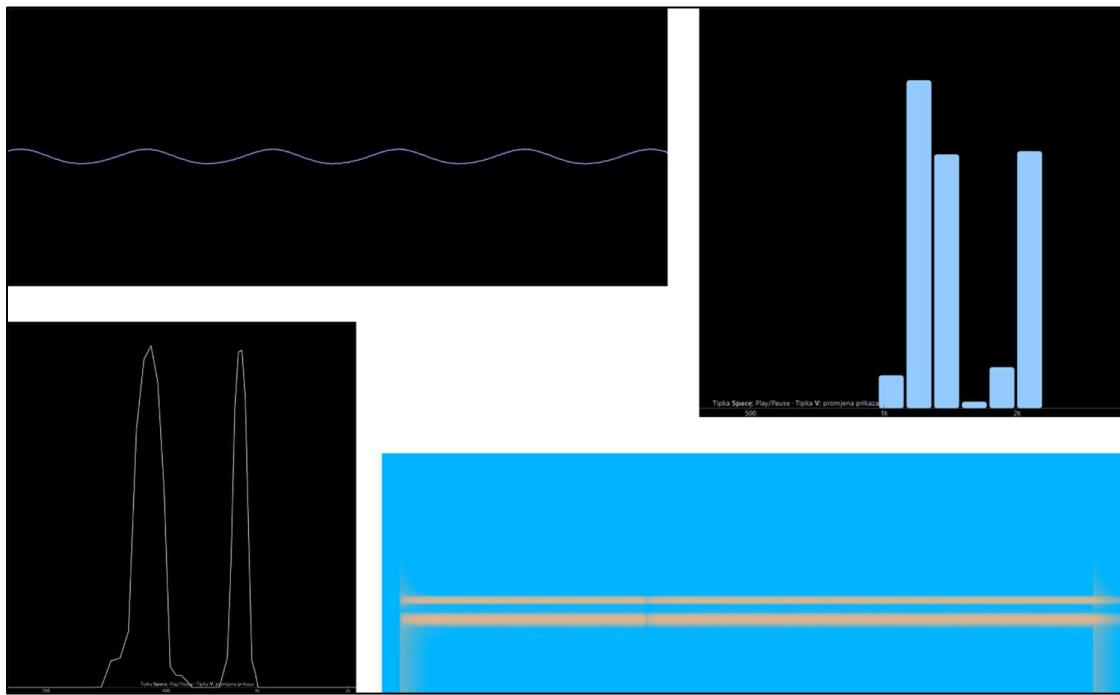
Slika 11 Rezultati vizualizacije – sinus (440 Hz)

#### 4.1.2. Dualni sinus (440 Hz i 880 Hz)

U tablici 2 bit će prikaz testiranja i evaluacije za testni signal dva sinusa (440 Hz i 880 Hz), a dobivene vizualizacije na slici 12.

Prikaz	Očekivanje	Uočeno
<b>Bars</b>	Dva jasna pojasa oko 440 Hz i 880 Hz	Dva jasna pojasa oko 440 Hz i 880 Hz <input checked="" type="checkbox"/>
<b>Line</b>	Dva uska vrha na 440 Hz i 880 Hz	Dva uska vrha na 440 Hz i 880 Hz <input checked="" type="checkbox"/>
<b>Spectrogram</b>	Dvije paralelne vodoravne linije	Dvije paralelne vodoravne linije <input checked="" type="checkbox"/>
<b>Scope</b>	Periodičan valni oblik	Periodičan valni oblik <input checked="" type="checkbox"/>

Tablica 2 Rezultati testiranja – dualni sinus (440 Hz i 880 Hz)



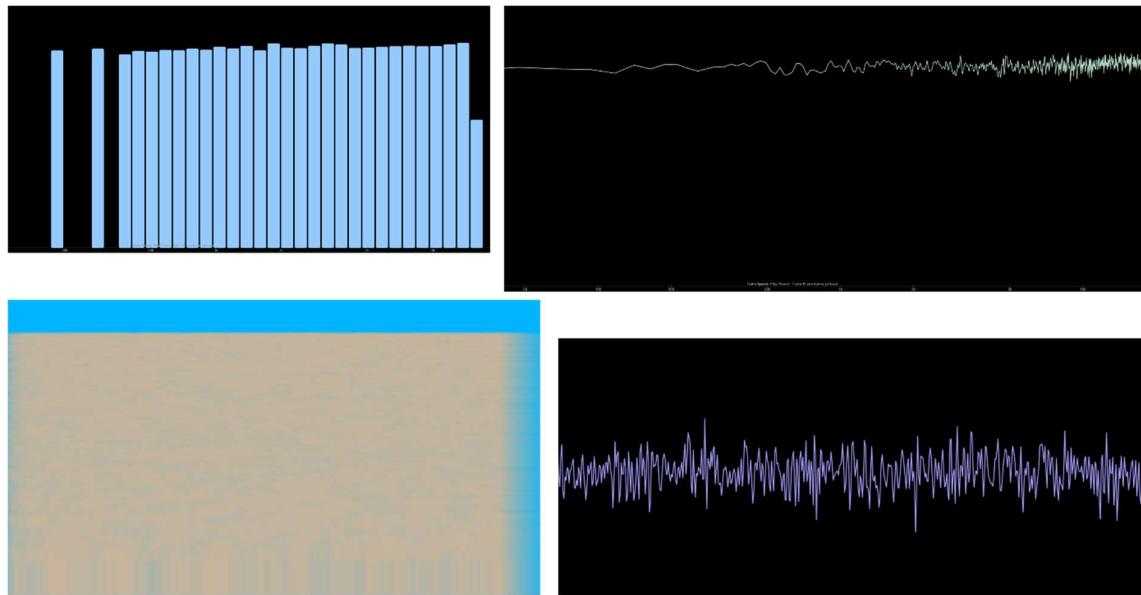
Slika 12 Rezultati vizualizacije – dualni sinus (440 Hz i 880 Hz)

#### 4.1.3. Bijeli šum

U tablici 3 bit će prikaz testiranja i evaluacije za testni signal bijeli šum, a dobivene vizualizacije na slici 13.

Prikaz	Očekivanje	Uočeno
<b>Bars</b>	Ravnomjeran spektar bez tonova	Ravnomjeran spektar bez tonova <input checked="" type="checkbox"/>
<b>Line</b>	“Zrnat” profil oko stalne razine	“Zrnat” profil oko stalne razine <input checked="" type="checkbox"/>
<b>Spectrogram</b>	Mozaik bez uzorka	Mozaik bez uzorka <input checked="" type="checkbox"/>
<b>Scope</b>	Neperiodičan signal oko nule	Neperiodičan signal oko nule <input checked="" type="checkbox"/>

Tablica 3 Rezultati testiranja – bijeli šum



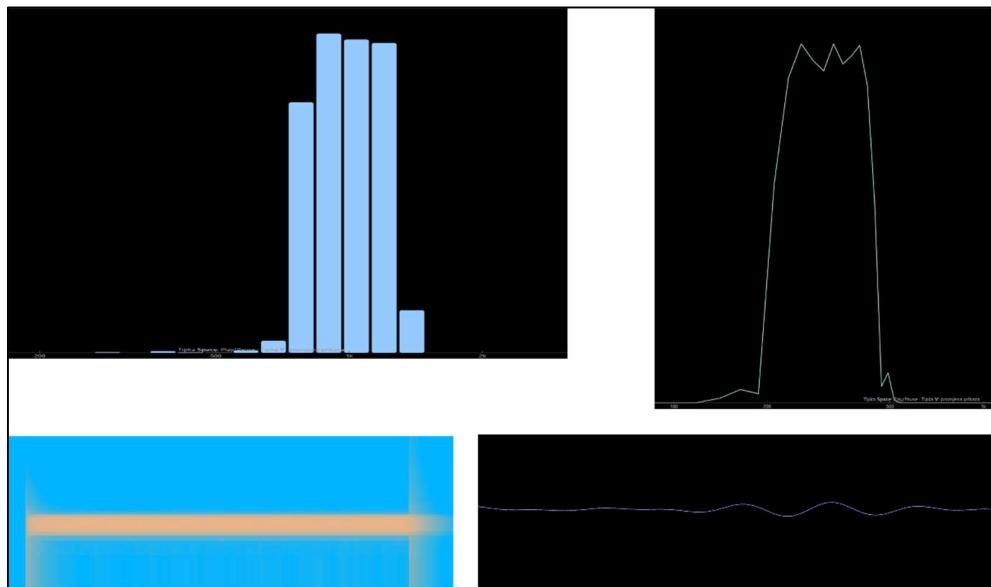
Slika 13 Rezultati vizualizacije – bijeli šum

#### 4.1.4. Akord C-E-G

U tablici 3 bit će prikaz testiranja i evaluacije za testni signal akord C-E-G, a dobivene vizualizacije na slici 14.

Prikaz	Očekivanje	Uočeno
<b>Bars</b>	3 stupca $\approx 262/330/392$ Hz	3 stupca $\approx 262/330/392$ Hz <span style="color: green;">✓</span>
<b>Line</b>	3 uska vrha + slabiji harmonici	3 uska vrha + slabiji harmonici <span style="color: green;">✓</span>
<b>Spectrogram</b>	3 paralelne vodoravne linije	3 paralelne vodoravne linije <span style="color: green;">✓</span>
<b>Scope</b>	Valni oblik složeniji od sinusoide	Valni oblik složeniji od sinusoide <span style="color: green;">✓</span>

Tablica 4 Rezultati testiranja – akord C-E-G



Slika 14 Rezultati vizualizacije – akord C-E-G

## 4.2. Evaluacija na različitim žanrovima glazbe

Ispitivanje i evaluacija će se provesti jednu pjesmu za svaki od tri različita žanra glazbe: EDM, klasična glazba i rock/metal. Bit će korišten *smoothing* 0,85 za sva testiranja.

### 4.2.1. EDM – „Swedish House Mafia ft. John Martin - Don't You Worry Child“

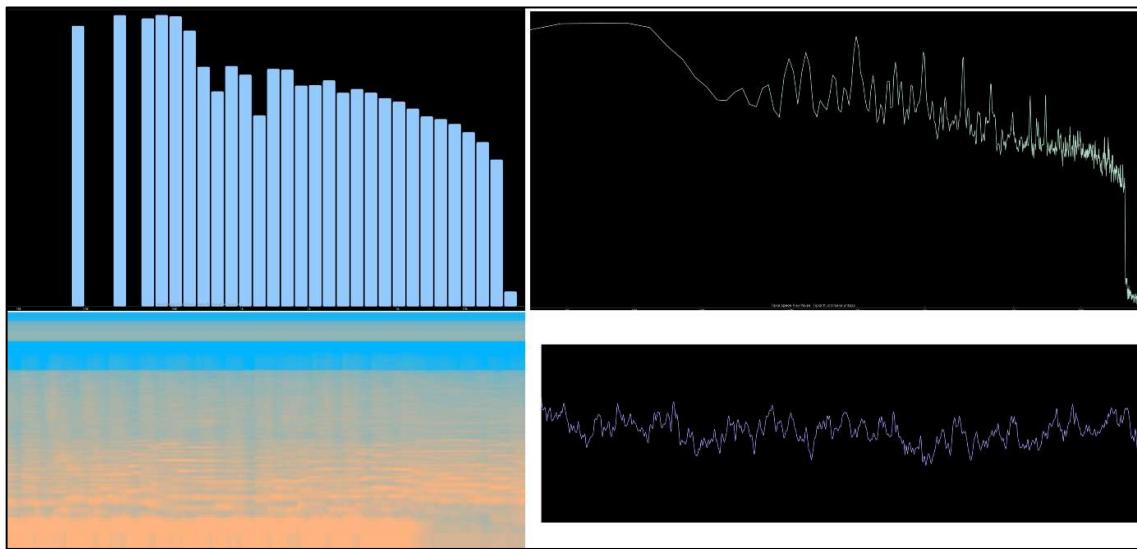
**Bars** – raspodjela energije nagnje prema niskim frekvencijama, najviši stupci su u području ~20 - 120 Hz (*sub-bas i kick*), potom slijedi stabilniji plato do oko 1 kHz i onda se vidi postupan pad jačine prema visokima. Ovakav oblik je tipičan za EDM gdje su najvažniji elementi ritam i sub-bas.

**Line** – linijski spektar pokazuje naglašeni niski pojas i nekoliko sitnih vrhova u srednjavisokom području. Krivulja ima očekivani blagi nagib prema dolje – prevladava ritmički sadržaj

**Spectrogram** – vidljive su razmaknute vertikalne trake koje odgovaraju udarcima *kicka* i *snares* uz kontinuiran niskofrekvencijski pojas pri dnu (*sub-bas*).

**Scope** – vremenski prikaz je pulsirajući, amplituda raste u dijelovima „*dropa*“.

Vizualizacija potvrđuje ono što je i očekivano; niska osnova i ritamski otisci u spektrogramu. Sustav dobro razlikuje i naglašava ključne elemente ovog žanra kao što se vidi na prikazima na slici 15.



Slika 15 Rezultati vizualizacije – EDM

#### 4.2.2. Klasična glazba – „Chopin: Nocturne op. 9 br. 2“

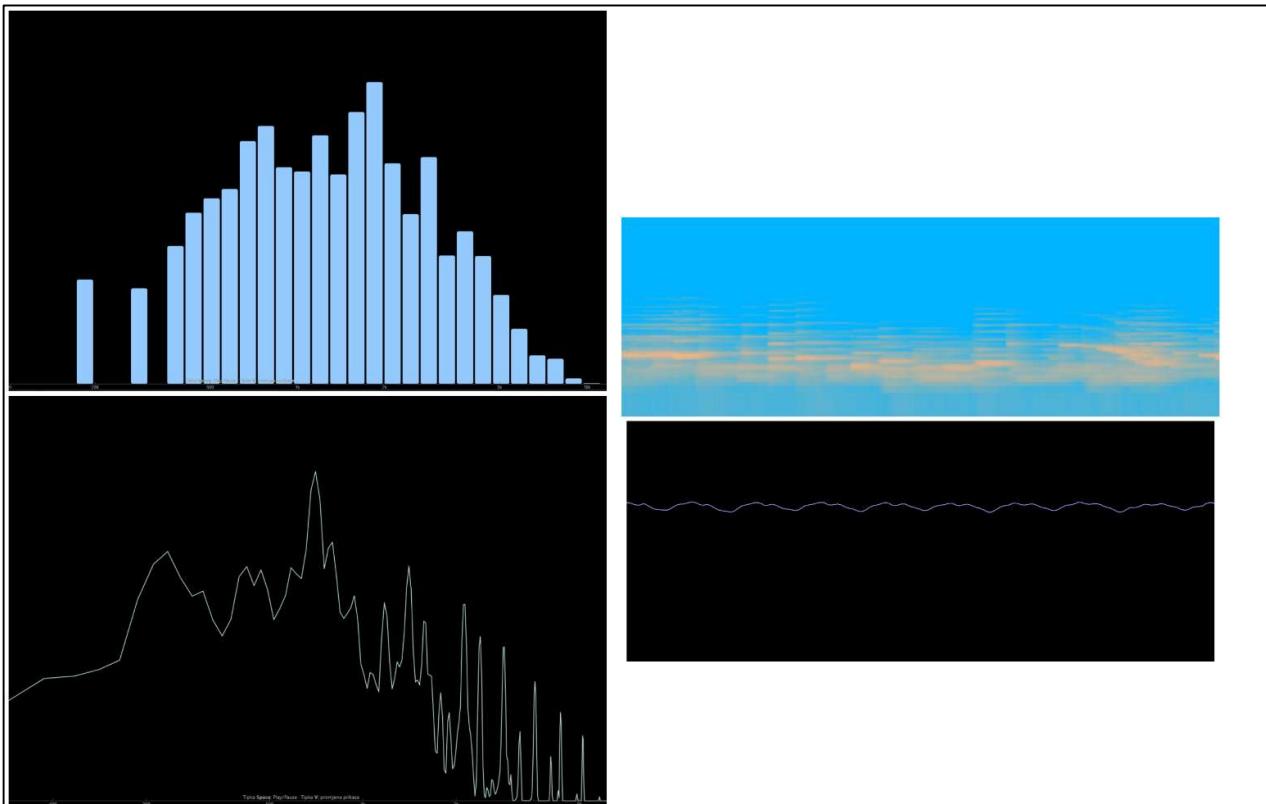
**Bars** - raspodjela energije je usmjerenja prema srednjim frekvencijama, najviši stupci nalaze se otprilike između 500 Hz i 3 kHz, što odgovara osnovnim tonovima klavira i njihovim nižim harmonikama. U bas području je energija skromnija, a iznad ~5–6 kHz primjetan je postupan pad.

**Line** - linijski spektar pokazuje više vrhova koji slijede harmonički niz pojedinih tonova. Vrhovi su relativno uski i stabilni što ukazuje na tonalno čist sadržaj.

**Spectrogram** - uočene su vodoravne, dugotrajne linije – fundamental i harmonici držanih tonova. Prijelazi između nota vide se kao blagi stepenasti pomaci linija; nema gustih vertikalnih pruga.

**Scope** - vremenski signal je mirniji i pravilniji nego kod modernih žanrova: dominira valni oblik koji ima male promjene amplitude

Vizualizacije naglašavaju čvrstu tonalnu strukturu i dugotrajanost tonova karakterističnu za klasičnu glazbu kao što se vidi na prikazima na slici 16. U spektrogramu dominiraju vodoravne linije, a vremenski prikaz ostaje ured i bez agresivnih udara.



Slika 16 Rezultati vizualizacije – klasična glazba

#### 4.2.3. Rock/metal – „System Of A Down: Chop Suey!“

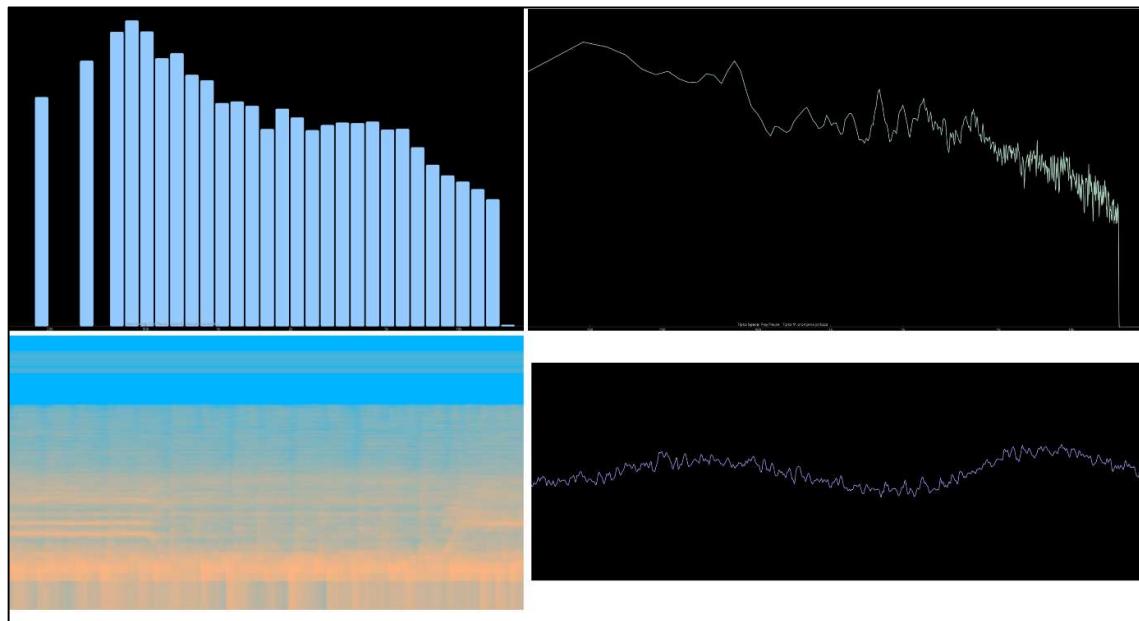
**Bars** – energetski profil je uravnotežen: iako su niske frekvencije prisutne zbog basa i kickom vrhunac je u srednjem području gdje dominiraju gitare i vokali.

**Line** – linijski spektar je gust s mnoštvom vrhova bez izoliranih tonova.

**Spectrogram** – slika je kontinuirano ispunjena u srednjem i višem području.

**Scope** – vremenski prikaz je kompleksan i kaotičan, nema čiste periodike, u skladu s više istovremenih izvora (gitare, bas, bubanj, vokal).

Vizualizacije precizno prikazuju zasićen, srednje-visoki spektar i prirodu rock/metala.



Slika 17 Rezultati vizualizacije – rock/metal

#### 4.3. Usporedba žanrova (EDM, Rock/Metal, Klasična)

EDM/Hip-hop pokazuje najveću koncentraciju energije u niskim frekvencijama (20–120 Hz); Bars ističe sub-bas i *kick*, dok Line nema stabilne tonalne vrhove nego blage “zubce”. Rock/Metal je uravnoteženiji, ali s naglašenim sredinama (0.5–3 kHz) zbog gitara i vokala; Line je gušći s mnoštvom harmonika, a Bars održava povišenu razinu kroz srednji pojas. Klasična glazba se razlikuje diskretnim tonalnim vrhovima i urednim harmonijskim nizovima; Bars pokazuje izolirane stupce, a Line jasno razlučuje harmonike uz slabiji sub-bas i prirodan pad prema visokim frekvencijama.

Na spektrogramu EDM dominiraju vertikalne pruge i kontinuirani niski pojasi; Rock/Metal je gusto ispunjen u srednje-visokom području; Klasična je prepoznatljiva po vodoravnim i dugotrajnim linijama i mirnijem vremenskom obrascu.

## 5. OGRANIČENJA I MOGUĆNOSTI NADOGRADNJE

Postoji nekoliko **ograničenja** koja je teško izbjegći prilikom vizualizacije zvuka. Vizualizator se u jezgri oslanja na p5.FFT s 1024 frekvencijskih *binova*. Razlučivost je time ograničena pa se vrlo bliski tonovi teško potpuno razdvajaju, osobito u „Bars“ prikazu. Log-mapiranje i diskretni prozor suočavaju se sa uobičajenim razmazivanjem spektra, a izlaz je u relativnim dB bez akustične kalibracije. Vremensko zaglađivanje (EMA) učinkovito smiruje grafove, ali uvodi odgodu reakcije reda nekoliko *frameova* pa se vrlo brze promjene često mogu previše ublažiti.

## 6. ZAKLJUČAK

U ovom radu razvijena je web stranica za vizualizaciju zvuka koji radi u stvarnom vremenu, koristeći p5.js i p5.sound, s pet komplementarnih prikaza: "Bars", "Line", "Spectrogram", "Scope" i "Circular". U sustav je dodan *slider* za zaglađivanje (EMA) koji omogućava korisnicima da biraju između dinamične i mirne vizualizacije, a kontrole putem tipkovnice (Space za reprodukciju/pauzu, V za promjenu prikaza) čine sučelje jednostavnim i efikasnim za korištenje.

Testiranje na kontroliranim signalima (sinus, dualni sinus, akord C–E–G i bijeli šum) pokazalo je da sustav točno prikazuje osnovne uzorke i očekivane efekte u vremenskoj i frekvencijskoj domeni. Evaluacija kroz tri glazbena žanra (EDM, Rock/Metal, Klasična glazba) dodatno je dokazala korisnost vizualizacija: ritmički elementi i sub-bas u EDM-u, bogate sredine i kontinuum u rock/metal pjesmama, te jasna tonalnost, mirnoća i vodoravne linije u klasičnoj glazbi.

Unatoč određenim metodološkim ograničenjima (razlučivost FFT analize, relativna dB-skala, inercija EMA-e, trošak CPU-a za spektrogram), predložene nadogradnje nude jasnou strategiju za postizanje preciznije i interaktivnije analize, a da pritom ne odstupaju od jednostavne web infrastrukture. Općenito, ostvareno rješenje je stabilno, pregledno i korisno za obrazovne svrhe: u stvarnom vremenu prikazuje ključne osobine zvuka, a ostaje dovoljno lagano za jednostavnu upotrebu i daljnje proširenje.

## LITERATURA

- [1] Wikipedia, “Discrete Fourier transform.” Dostupno na:  
[https://en.wikipedia.org/wiki/Discrete\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform) (Pristupljeno: 15. 9. 2025.).
- [2] Buck, J. R., Oppenheim, A. V., Schafer, R. W. (1999). *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall
- [3] Cooley, J. W. (1987). “The re-discovery of the fast Fourier transform algorithm,” *Mikrochimica Acta*, vol. 93, pp. 33–45
- [4] NTi Audio, “Fast Fourier Transform (FFT) – Know-How.” Dostupno na: <https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft> (Pristupljeno: 15. 9. 2025.).
- [5] Allen, J. B., Rabiner, L. R. (1977). “A unified approach to short-time Fourier analysis and synthesis,” *Proceedings of the IEEE*
- [6] Smith III, J. O. (2011). *Spectral Audio Signal Processing*. Stanford, CA, 2011.
- [7] Moore, B. C. J. (2012). *An Introduction to the Psychology of Hearing*, 6th ed. Bingley, UK: Emerald Group Publishing
- [8] Lyons, R. G. (2011). *Understanding Digital Signal Processing*, Pearson, NJ
- [9] Zovko P. (2022). *Razvoj i implementacija digitalnog filtra*, Sveučilište u Splitu – Sveučilišni odjel za stručne studije. Dostupno na:  
<https://repositorij.oss.unist.hr/islandora/object/ossst%3A1775> (Pristupljeno: 15.9.2025.).