

Kodovanje zvučnog i video signala

Predmet: Multimedijalni signali i sistemi

Predavač: dr Nataša Savić

Asistent: Nikola Milutinović

Kodovanje audio signala

%% 1. Parametri signala

```
fs = 44100;      % Frekvencija samplinga (CD kvalitet)
t = 0:1/fs:2;   % Trajanje 2 sekunde
```

%% 2. Generisanje sintetičkog audio signala

```
f1 = 440;      % A4 ton
f2 = 660;      % Dodatna harmonija
f3 = 880;      % Još jedna komponenta
audioSignal = 0.5*sin(2*pi*f1*t) + ...
              0.3*sin(2*pi*f2*t) + ...
              0.2*sin(2*pi*f3*t);
```

%% 3. Normalizacija signala

```
audioSignal = audioSignal / max(abs(audioSignal));
```

%% 4. PCM Kvantizacija (kodovanje)

```
nBits = 8;      % Broj bitova po sample-u
L = 2^nBits;    % Broj nivoa
xmin = -1; xmax = 1; % Opseg signala
```

% Uniformna kvantizacija

```
delta = (xmax - xmin)/L;      % Korak kvantizacije
qSignal = floor((audioSignal - xmin)/delta); % Kvantizovani indeks
```

% Ograničavanje indeksa

```
qSignal(qSignal > L-1) = L-1;
qSignal(qSignal < 0) = 0;
```

% Rekonstrukcija signala (dekvantizacija)

```
recSignal = (qSignal * delta) + xmin + delta/2;
```

%% 5. Prikaz rezultata

```
figure;
subplot(3,1,1);
plot(t(1:2000), audioSignal(1:2000));
title('Originalni audio signal (prvih 2000 uzoraka)');
xlabel('t [s]');
ylabel('Amplituda');
```

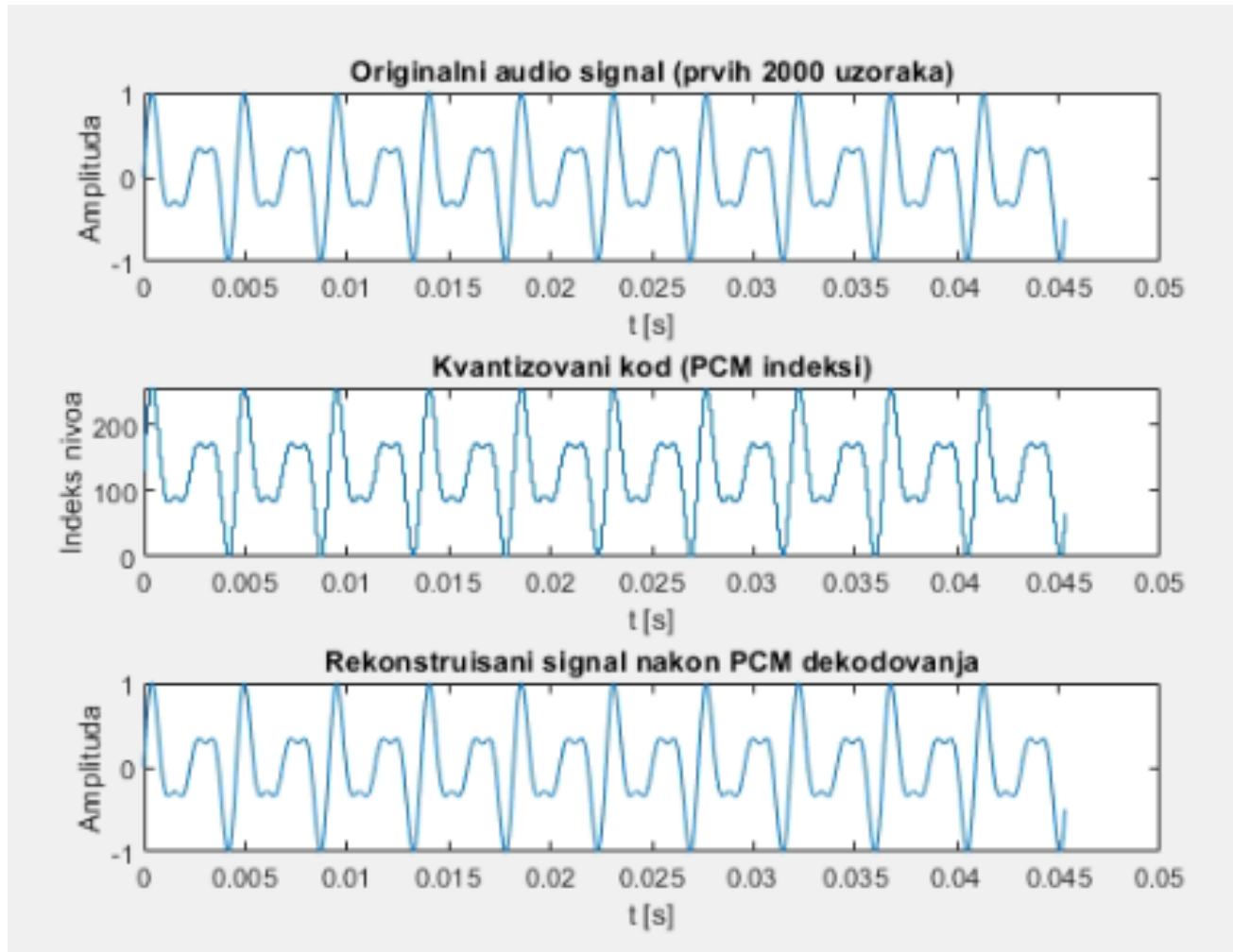
```
subplot(3,1,2);
stairs(t(1:2000), qSignal(1:2000));
title('Kvantizovani kod (PCM indeksi)');
xlabel('t [s]');
ylabel('Indeks nivoa');
```

```
subplot(3,1,3);
plot(t(1:2000), recSignal(1:2000));
title('Rekonstruisani signal nakon PCM dekodovanja');
xlabel('t [s]');
ylabel('Amplituda');
```

%% 6. Čuvanje signala u .wav format (opciono)

```
audiowrite('synthetic_audio.wav', audioSignal, fs);
disp('Audio signal je sačuvan kao synthetic_audio.wav')
```

Rezultati



Kodovanje video signala

%% 1. Parametri video signala

```

fps = 30;
trajanje = 3;      % Kraće trajanje da se brže testira
N = fps * trajanje;
sirina = 480;
visina = 360;
    
```

%% 2. VideoWriter – kodovanje

```

videoObj = VideoWriter('encoded_video.avi');
videoObj.FrameRate = fps;
open(videoObj);
    
```

%% 3. Generisanje sintetičkog videa

```

radius = 30;
x_pos = linspace(40, sirina-40, N);
y_pos = visina/2;
original_frames = cell(1,N);
for k = 1:N
    % Prazan frejm
    frame = zeros(visina, sirina, 3, 'uint8');
    
```

% Pozicija kruga

```

x = round(x_pos(k));
y = round(y_pos);
    
```

% Crtanje kruga

```

for i = -radius:radius
    for j = -radius:radius
        if i^2 + j^2 <= radius^2
            if y+i > 0 && y+i <= visina && x+j > 0 && x+j <= sirina
                frame(y+i, x+j, 1) = 255;
            end
        end
    end
end
    
```

% Čuvanje originalnog frejma za kasniju analizu

```

original_frames{k} = frame;
    
```

% Pisanje u video (kodovani video)

```

writeVideo(videoObj, frame);
end
close(videoObj);
disp('Video signal je generisan i kodovan.');
```

%% 4. Učitavanje kodiranog videa i rekonstrukcija

```

vid = VideoReader('encoded_video.avi');
decoded_frames = cell(1, N);
k = 1;
while hasFrame(vid)
    decoded_frames{k} = readFrame(vid);
    k = k + 1;
end
disp('Kodirani video je učitani.');
```

Kodovanje video signala

% 5. Prikaz originalnog vs. kodiranog signala

sample_frame = round(N/2); % Uzmi srednji frejm za poređenje

org = original_frames{sample_frame};

dec = decoded_frames{sample_frame};

diff_frame = imabsdiff(org, dec); % razlika između signala

figure;

subplot(1,3,1);

imshow(org);

title('Originalni video frejm');

subplot(1,3,2);

imshow(dec);

title('Kodirani (dekodirani) video frejm');

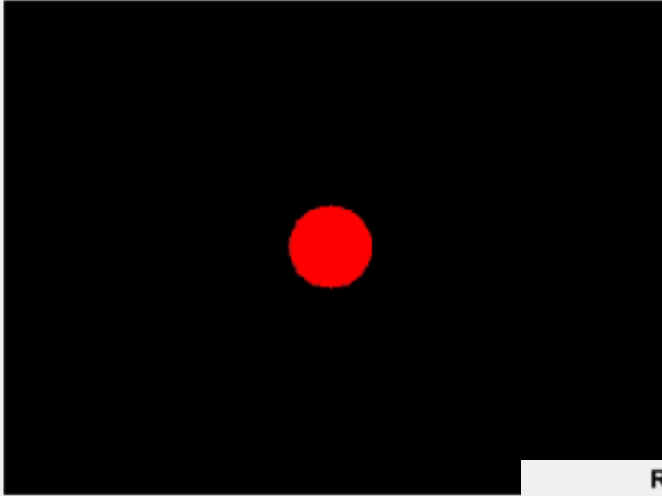
subplot(1,3,3);

imshow(diff_frame .* 5); % pojačanje da bude vidljivo

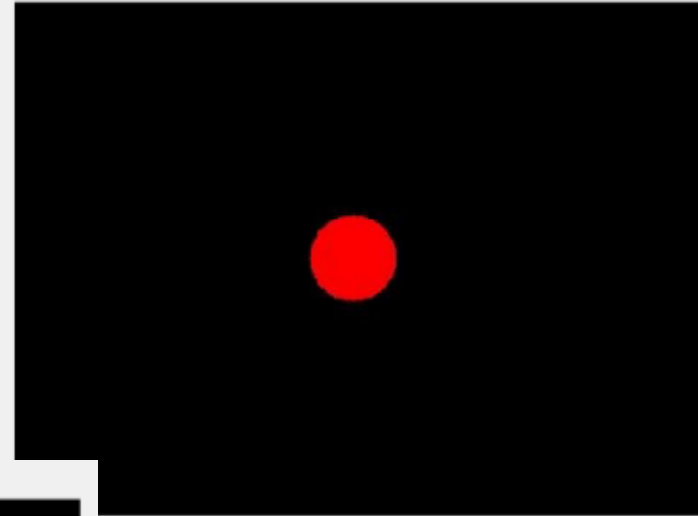
title('Razlika između originalnog i kodiranog');

Rezultati

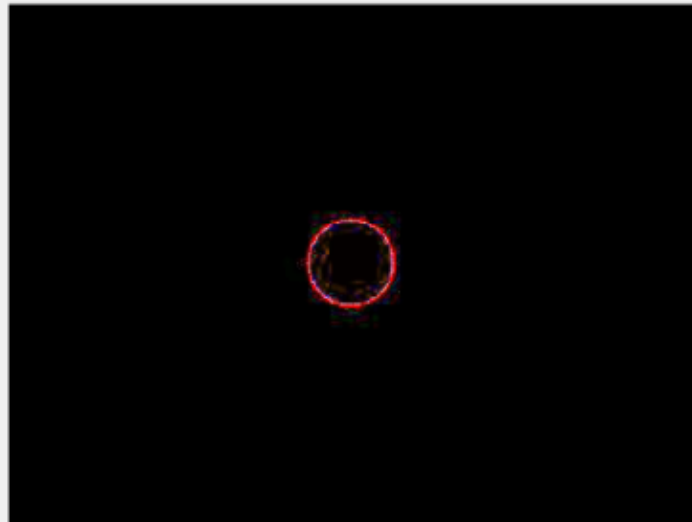
Originalni video frejm



Kodirani (dekodirani) video frejm



Razlika između originalnog i kodiranog



Hvala na pažnji!

PITANJA?

