

## Vježba 1.

# Predstavljanje multimedijalnih signala u MATLAB-u

### **Uvod u MATLAB**

MATLAB je jezik za naučna i inženjerska izračunavanja. Počevši od ove osnovne funkcionalnosti MATLAB je razvijen u sistem koji pored izračunavanja omogućava i vizuelizaciju, programiranje, povezivanje sa programima napisanim u Javi, C-u ili Fortranu. Posebnu snagu MATLAB-u daju biblioteke funkcija specijalizovane za rješavanje pojedinih problema (toolboxes). Funkcionalnosti MATLAB-a se može pristupiti korištenjem grafičkog integrisanog razvojnog okruženja.

Prednosti korištenja MATLAB-a su:

- jednostavna sintaksa – lako izvršavanje numeričkih izračunavanja,
- jednostavna vizuelizacija rezultata,
- optimizovan za rad sa matricama i vektorima,
- na raspolaganju su brojne biblioteke za rješavanje praktičnih problema,
- podrška različitim multimedijalnim formatima,
- grafičko integrisano razvojno okruženje.

Sa druge strane, loše strane MATLAB-a se uglavnom odnose na činjenicu da je MATLAB interpretiran jezik što može rezultirati sporim izvršavanjem programa. Srećom, dobrom programerskom praksom se ovo u velikoj mjeri može izbjeći čime se dobija kraće *ukupno* vrijeme razvoja. Druga loša strana MATLAB-a je njegova cijena.

### **Uvodne napomene, vektori, matrice**

MATLAB se pokreće biranjem **Start** → **Programs** → **<oznaka verzije>** → **Matlab**. Nakon ovoga otvara se komandni prozor i pojavljuje MATLAB prompt `>>`.

Iz MATLAB-a se može izaći kucanjem komande:

```
»quit
```

U MATLAB je ugrađen i obiman sistem pomoći koji se dobija komandom:

```
»help
```

Odnosno, ako znate ime funkcije, ali se ne možete sjetiti tačne sintakse:

```
»help ime_funkcije
```

Većina matematičkih funkcija je ugrađena u MATLAB i obično je korisno da se komanda `help` izvrši za željenu funkciju da bi se upoznali sa njenim mogućnostima.

Osnovna struktura podataka u MATLAB-u je matrica. Specijalni slučajevi su  $1 \times 1$  matrica (skalar) i matrice čija je jedna dimenzija 1 – vektori.

Matrice se mogu unijeti:

1. nabranjem svih elemenata,
2. učitavanjem matrice iz fajla,
3. generisanje pomoću ugrađenih funkcija,
4. generisanje pomoću sopstvenih (korisničkih) funkcija.

Na primjer, sledećim naredbama se definišu skalar, vektor i matrica, respektivno.

```
»x = 5
»b = [1 3 8 12.1 7]
»A = [1, 7, 8; 2, 9, 13; 4, 8, 11; 2, 5, 3]
```

MATLAB nakon izvršenja svake komande ispisuje rezultat. To se može izbjeći ako se linija završi sa ;

Koristan način za zadavanje vektora može biti operator dvotačka : čija je sintaksa

```
x = start:korak:kraj
```

Ukoliko korak nije zadat podrazumijeva se da je njegova vrijednost jednaka 1. Na primjer:

```
» x = 3:8
```

```
x =
```

```
3 4 5 6 7 8
```

```
» x = 2:2:10
```

```
x =
```

```
2 4 6 8 10
```

Korak može imati i negativnu vrijednost u kom slučaju se dobijaju vrijednosti u vektoru u opadajućem redosljedu:

```
» x = 10:-2:2
```

```
x =
```

```
10 8 6 4 2
```

Nijedna od vrijednosti u ovom operatoru ne mora biti cjelobrojna:

```
» x = 0:0.2:1
```

```
x =
```

```
0 0.2 0.4 0.6 0.8 1
```

Pojedinim elementima matrice može se pristupiti navođenjem njihovih indeksa:

```
» b(1)
```

```
ans =
```

```
1
```

```
» A(1,2) = 1
```

Indeksi ne moraju biti skalari, umjesto njih mogu se upotrebiti proizvoljni cjelobrojni vektori. Na ovaj način se iz date matrice mogu izdvojiti podmatrice. Npr. sledeća naredba daje matricu u kojoj su vrste druga i četvrta vrsta matrice A, a kolone prva i treća kolona iste matrice:

```
>> A([2 4], [1 3])
```

```
ans =
```

```
2 13
```

```
2 3
```

Vrlo često se za kreiranje indeksa koristi operator dvotačka. Npr. sledeća naredba vraća vektor-kolonu koji se sastoji od prva tri elementa prve kolone matrice A.

```
>> A(1:3, 1)
```

```
ans =
```

```
1
```

2

4

Ako se zada samo dvotačka onda se operacija izvršava nad cijelom vrstom ili kolonom:

```
» A(1,:) = [1, 7, 8]
```

```
» A(:,1) = [1; 2; 4; 2]
```

Vještom upotrebom operatora dvotačka moguće je izbjeći korištenje for petlji prilikom programiranja u MATLAB-u čime se značajno poboljšavaju performanse programa.

Postoji i niz funkcija koje generišu karakteristične matrice. Onovne su

```
» A = zeros(n,m);
```

```
» B = ones(n,m);
```

```
» C = eye(n,m);
```

```
» D = rand(n,m);
```

Zeros generiše matricu dimenzija (n,m) čiji su svi elementi nule, ones matricu istih dimenzija čiji su svi elementi jedinice, eye jediničnu matricu, a rand matricu zadatih dimenzija čiji su elementi slučajni brojevi iz uniformne raspodjele na intervalu [0, 1]. Ovo je vrlo korisno kod zadavanja signala.

Stringovi se u MATLAB-u zadaju korištenjem apostrofa ('), npr. s = 'string'. Interno, string se memoriše kao vektor čiji su elementi pojedini karakteri.

Pored promjenljivih definisanih na pomenute načine u MATLAB-u postoje i neke specijalne predefinisane promjenljive:

| Specijalne promjenljive | Opis                                                                                      |
|-------------------------|-------------------------------------------------------------------------------------------|
| ans                     | Rezultat poslednje operacije                                                              |
| pi                      | Broj $\pi$                                                                                |
| eps                     | Najmanji broj koji kada se doda jedinici, daje broj koji je na računaru veći od jedinice. |
| flops                   | Broj floating-point operacija                                                             |
| inf                     | Beskonačno (npr. 1/0).                                                                    |
| NaN (or) nan            | Nije broj – Not-a-Number (npr. 0/0).                                                      |
| i (and) j               | $i = j = \sqrt{-1}$                                                                       |
| realmin                 | Najmanji mogući realan broj                                                               |
| realmax                 | Najveći upotrebljiv pozitivan realni broj                                                 |

## Osnovne operacije i funkcije nad matricama

Osnovne aritmetičke operacije +, -, \*, / mogu da se koriste na isti način kao i u drugim višim programskim jezicima s tim što njihovi argumenti mogu biti i vektori, odnosno matrice. Na primjer ukoliko želite da pomnožite matrice A i B dovoljno je upotrebiti naredbu: C = A \* B.

Isto vrijedi i za elementarne funkcije: `exp`, `log`, `sqrt`, `sin`, `cos` itd. Ako su argumenti ovih funkcija matrice onda se elementarne funkcije izračunavaju po elementima.

Dimenzije matrice dobijaju se pomoću funkcije `size`.

```
» size(A)
```

Transponovanje niza/matrice vrši se pomoću operatora `'`

```
» C = C'
```

```
C =  
    1  
    2  
    3  
    4  
    5  
    6  
    7  
    8
```

Da biste izvršili skalarnu operaciju na vektoru koristite obične matematičke operacije. Na primjer, da biste pomnožili svaki element vektora `C` sa 2 koristite komandu:

```
» C = 2*C
```

```
C =  
     2     4     6     8    10    12    14    16
```

Pretpostavimo da želite da pomnožite svaki element vektora `a` odgovarajućim elementom vektora `b`. U tom slučaju koristićete operator `.` koji izvršava operacije po elementima. Na primjer:

```
» a = [2 2 2 2]
```

```
a =  
     2     2     2     2
```

```
» b = [1 2 3 4]
```

```
b =  
     1     2     3     4
```

```
» a.*b
```

```
ans =  
     2     4     6     8
```

Ako biste koristili obično množenje `*` dobili biste:

```
» a*b
```

```
??? Error using ==> *
```

```
Inner matrix dimensions must agree.
```

zato što MATLAB pokušava da pomnoži matrice `a` i `b`, prema pravilima za množenje matrica.

Konkatenacija nizova je takođe veoma jednostavna:

```
>> A = [1 2 3 4];
```

```
>> B = [5 6 7 8];
```

```
» C = [A B]
```

```
C =  
    [1 2 3 4 5 6 7 8]
```

## Grafičko prikazivanje podataka

Osnovna funkcija za grafički prikaz podataka je `plot`. U zavisnosti od ulaznih argumenata dobijaju se različiti rezultati. Na primjer,

```
» x = [1, 7, 5, 4.3, 2, 9, 11, 8.8];  
» plot(x)
```

će nacrtati tačke iz vektora `x` u funkciji njihovog indeksa i povezati ih pravolinijskim segmentima, a

```
» plot(x, 'o')
```

će nacrtati tačke označene kružićima i neće ih povezati.

## Primjer

Nacrtati funkciju  $y = x^2$  na intervalu  $[-2, 2]$ .

```
x = -2:0.01:2;
```

```
y = x.^2;
```

```
plot(x,y);
```

Moguće je na jednoj slici nacrtati više grafika i MATLAB ih crta različitim bojama iz predefinisano skupa.

```
y1 = x.^2 - 4;
```

```
y2 = x.^2 + 4;
```

```
plot(x,y,x,y1,x,y2);
```

Boja, oblik tačkica, tip linija se mogu zadati eksplicitno, pogledati help stranicu za komandu `plot`.

Komanda `plot` automatski otvara novi prozor, ako ne postoji ni jedan već otvoren. Ako je neki grafički prozor otvoren, `plot` crta u njega, a prethodni sadržaj se briše. Novi prozor se otvara pomoću komande `figure` i on postaje aktivan. Sledeća `plot` komanda crta u njega. Već postojeći prozor se aktivira sa `figure(n)`, gdje je `n` broj prozora koji se nalazi u naslovnoj liniji, npr. Figure No.1

Pomenuto je da se crtanjem u već postojeći prozor njegov prethodni sadržaj briše.

Ovakvo se ponašanje može promijeniti pomoću komande

```
>> hold on
```

koja zadržava postojeći grafik i sledeća komanda za crtanje crta preko njega.

Predefinisano ponašanje vraća se sa:

```
>> hold off
```

Graficima se lako mogu dodati naslov, kao i oznake osa.

```
t = -pi:pi/100:pi;
```

```
y = sin(t);
```

```
plot(t,y);
```

```
title('Grafik funkcije sinus');
```

```
xlabel('t');
```

```
ylabel('sin(t)');
```

### 3D grafika

MATLAB posjeduje i veoma napredne mogućnosti za 3D inženjersku grafiku. Osnovne naredbe za generisanje 3D grafika su `surf` i `mesh`. Kao demonstracija može da posluži sledeća sekvenca naredbi:

```
surf(40*membrane(1,25));  
shading interp;  
daspect([1 1 0.9]);  
camlight  
axis off tight vis3d;
```

Iz linije sa alatima izaberi Rotate 3D koji omogućava rotiranje prikazanog objekta klikom i pomjeranjem miša.

### Predstavljanje multimedijalnih signala u MATLAB-u

Tema ove vježbe je upoznavanje sa načinom predstavljanja multimedijalnih signala u MATLAB-u. Kao i u drugim programskim okruženjima, u MATLAB-u su signali predstavljeni kao nizovi, odnosno, matrice.

### Zvučni signal

Zvučni signal se u MATLAB-u predstavlja kao vektor čiji su elementi odmjerci signala. MATLAB može čitati WAV i AU muzičke fajlove korištenjem funkcija `wavread()` i `auread()`, respektivno. Npr. naredba:

```
[y, Fs, nbits] = wavread('handel.wav');
```

učitava zvučni signal u datom WAV fajlu u vektor `y`. U promjenljivoj `Fs` će se nalaziti frekvencija odmjjeravanja, a u promjenljivoj `nbits` broj bita korišten za kodiranje signala.

Najjednostavnija naredba kojom se može reprodukovati zvuk je `sound()`. Njena sintaksa je:

```
sound(y, Fs)
```

Reprodukcija pomoću naredbe `sound()` je asinhrona. Više mogućnosti za kontrolu reprodukcije pruža korištenje `audioplayer` objekta. Konstruktoru objekta se prosljeđuju: vektor odmjjeraka, frekvencija odmjjeravanja i (opciono) broj bita korišten za kodiranje signala:

```
p = audioplayer(y, Fs)
```

Asinhrona reprodukcija se sada može postići korištenjem metoda `play()`:

```
play(p)
```

Reprodukcija se može zaustaviti:

```
stop(p)
```

ili pauzirati:

```
pause(p)
```

Sinhrona reprodukcija se postiže korištenjem metoda `playblocking()`:

`playblocking(p)`

Svi metodi objekta se mogu dobiti pomoću:

`methods(p)`

Svi metodi klase `audioplayer` se mogu dobiti pomoću:

`methods audioplayer`

Slično, osobine objekta, odnosno, klase se mogu dobiti pozivom naredbe `properties`, a događaji pozivom naredbe `events`.

Dokumentacija za konkretan metod ili osobinu može se dobiti pomoću

`help <ime klase>/<ime metode ili osobine>`

MATLAB može i vrijednosti vektora zapisi u WAV ili AU formatu korištenjem funkcija `wavwrite()` i `auwrite()`. Npr.

`wavwrite(y, Fs, nbits, filename)`

upisuje u fajl `filename` elemente vektora `y` sa frekvencijom odmjeravanja `Fs` i `nbits` za kodiranje odmjeraka.

## Slika

Slika se u MATLAB-u predstavlja kao dvodimenzionalna ili trodimenzionalna matrica čiji elementi određuju boju piksela na određenoj lokaciji. Ukoliko se radi o grayscale slici matrica je dvodimenzionalna i vrijednosti elemenata matrice predstavljaju intenzitet piksela. Ukoliko se radi o slici u boji matrica je trodimenzionalna i sadrži intenzitete piksela za svaki od tri kolor kanala: R, G i B.

Osnovne informacije o slici koja se nalazi u nekom fajlu na disku moguće je dobiti korištenjem funkcije `imfinfo()`.

Slika se u memoriju učitava pomoću funkcije `imread()`. Na primjer, ako želimo da učitamo sliku `lenacolor.jpg` možemo iskoristiti sledeću naredbu:

```
slika = imread('lenacolor.jpg');
```

Funkcijom `imread()` se mogu učitati slike u većini standardnih formata kao što su: JPEG, GIF, PNG, TIFF, itd.

Slika koja se nalazi u matrici u radnom prostoru MATLAB-a se može prikazati pomoću funkcije `imshow()`, npr.

```
imshow(slika).
```

Pored ove funkcije na raspolaganju je i sofisticiraniji interaktivni alat za prikazivanje slika kojem se pristupa korištenjem funkcije `imtool()`:

```
imtool(slika).
```

Slika koja se nalazi u matrici u radnom prostoru MATLAB-a se može upisati u fajl na disku korištenjem funkcije `imwrite()`. Ova funkcija ima veliki broj opcija pomoću kojih se podešavaju različiti parametri fajlova. Najjednostavniji način pozivanja je:

```
imwrite(slika, 'lena.tiff').
```

## Video

Video klipovi se u MATLAB-u predstavljaju nizom čiji je svaki element struktura tipa `film` (movie) koja ima dva polja `cdata` i `colormap`. Broj elemenata strukture jednak je

broju frejmova u videu. U polju `cdata` nalazi se slika koja odgovara datom frejmu, a u polju `colormap` odgovarajuća kolor-mapa. Polje `colormap` može biti i prazno u kom slučaju se radi o intenzitetskim slikama.

Osnovne informacije o video-klipu koji se nalazi u nekom AVI fajlu na disku se mogu dobiti korištenjem funkcije `aviinfo()`.

Video se u memoriju učitava korištenjem funkcije `aviread()`, npr.

```
m = aviread('FishTank.avi');
```

Reprodukcija filma koji se nalazi u strukturi `m` se inicira korištenjem naredbe `movie()`:

```
movie(m, n, fps),
```

gdje je `n` broj ponavljanja klipa, a `fps` broj frejmova u sekundi za reprodukciju.

Novi AVI fajl se može kreirati od MATLAB-ove promjenljive tipa film korištenjem funkcije `movie2avi`, npr:

```
movie2avi(m, 'ft.avi')
```

## Zadaci

### Zvučni signal

1. Učitati zvučni signal u fajlu `handel44100.wav` u radni prostor MATLAB-a. Kakvom strukturom podataka je zvučni signal predstavljen u memoriji? Kolika je frekvencija odmjerenja ovog signala i sa koliko bita je kodovan svaki odmjerek signala? Kolike su dimenzije vektora u kojem su odmjerci signala i koliko memorije zauzima? Koliko je trajanje signala u sekundama? Nacrtati signal korištenjem naredbe `plot`, kao da se radi o analognom signalu. Na apscisi označiti vrijeme u sekundama.
2. Zašto je frekvencija odmjerenja signala na CD-u 44100Hz? Znajući frekvenciju odmjerenja i broj bitova za kodovanje izračunati bit-rate zvučnog signala CD kvalitete, tj. koliko kilobajta podataka sadrži jedna sekunda muzike na CD-u. Koliki je memorijski prostor potreban za 74 minute muzike? Koliki je dinamički opseg signala na CD-u?
3. Reprodukovati zvučni signal iz tačke 1. Zašto je prilikom reprodukcije zvučnog signala potrebno zadati frekvenciju odmjerenja? Šta bi se desilo ako biste koristili nižu ili višu frekvenciju odmjerenja? Pokušajte upotrebiti npr. 22050Hz ili 88200Hz. Komentarisati rezultate.
4. Učitati i reprodukovati zvučne signale u fajlovima `handel22050.wav` i `handel11025.wav`. Uporediti kvalitetu reprodukcije ova tri signala i komentarisati razlike. U kakvoj vezi je frekvencija odmjerenja signala sa razlikom u kvaliteti?
5. U sljedećim tačkama radićemo sa zvučnim signalom iz tačke 1 (`handel44100.wav`). Reprodukovati dati zvučni signal unazad.
6. Vrijednosti odmjeraka signala vezane su za intenzitet zvučnog signala. Na primjer, množenjem signala nekom konstantom dobija se njegovo pojačanje (ili slabljenje). Generišite novi signal koji je pojačan 2 puta u odnosu na signal iz tačke 1. Obratite pažnju na to da će ukoliko amplituda signala pređe vrijednost  $\pm 1$  doći do njegovog odsijecanja. Reprodukujte dobijeni signal.



- Generišite vektor odmjeraka eksponencijalnog signala  $x(t) = e^{-t/T}$ , iste dužine kao dati zvučni signal. Vremenska promjenljiva  $t$  treba da uzima vrijednosti sa korakom  $1/F_s$ , gdje je  $F_s$  frekvencija odmjeravanja zvučnog signala. Pomnožite ga sa datim signalom i poslušajte rezultat. Zadatak ponovite za vrijednosti  $T \in \{1, 2, 3\}$  sekunde. Komentarisati dobijene rezultate. Na ovaj način se dobija fade-out efekat.
- Sačuvati jedan od rezultata iz prethodne tačke kao novi WAV fajl.

## Slika

- Pomoću `imfinfo` provjerite osnovne karakteristike slike `LenaRGB.tif`. Učitajte sliku korišćenjem funkcije `imread`. Kakvom strukturom podataka je slika predstavljena u memoriji? Kolike su dimenzije dobijene matrice? Pogledajte vrijednosti elemenata matrice dobijene na taj način. Kojem opsegu pripadaju njihove vrijednosti? Kojoj memorijskoj klasi (tipu) podataka pripadaju?
- Funkcija `imshow` prikazuje sliku koja se nalazi u matrici u radnom prostoru MATLAB-a na ekranu. Proučite sintaksu funkcije `imshow`. Koje tipove slika podržava funkcija `imshow`? Prikažite sliku `LenaRGB.tif` na ekranu.
- Pojedine komponente RGB slike moguće je izdvojiti i obrađivati kao zasebne intenzitetske slike. Neka se npr. RGB slika nalazi u matrici `a`. Sada je R komponentu moguće izdvojiti korištenjem `a(:, :, 1)` itd. Izdvojiti i prikazati u posebnim figure prozorima sve tri komponente slike `LenaRGB.tif` kao grayscale slike.
- Ukoliko želite da prikazete RGB sliku koja sadrži samo jednu od komponenata slike, kreirajte novu sliku u kojoj ćete ostale dvije komponente postaviti na nulu, npr.

```
r = a; % sačuvati originalnu sliku
r(:, :, 2:3) = 0; % komponente 2 i 3 (G i B)
```

Prikazati na ovaj način sve tri komponente slike `LenaRGB.tif`.

- Funkcijom `rgb2gray()` se RGB slika može konvertovati u grayscale sliku. Primijenite ovu funkciju na matricu slike `LenaRGB.tif` i rezultat sačuvajte u novoj matrici. Prikažite rezultat. Koje su dimenzije nove matrice. Objasnite razliku u odnosu na dimenzije matrice originalne slike.
- Invertovati grayscale sliku iz prethodne tačke i prikazati rezultat.
- Sačuvati sliku iz prethodne tačke u JPEG formatu.

## Video

- Pomoću funkcije `aviinfo()` ispitati osnovne karakteristike video klipa `mobile.avi`. Učitati video `mobile.avi`. Kojom strukturom podataka je video predstavljen u memoriji? Šta se nalazi u pojedinim poljima strukture? Kolike su dimenzije matrica? Kojoj memorijskoj klasi pripadaju njihovi elementi?
- Reprodukovati video u MATLAB-u.

3. Izdvojite pojedine frejmove videa kao slike. Kreirati negativ videa invertovanjem slika kao u tački 6. iz prethodnog odjeljka.
4. Sačuvati video iz prethodne tačke kao novi AVI fajl.