

OBRADA ZVUČNIH SIGNALA PRIMJENOM PROGRAMSKOG JEZIKA MATLAB

AUDIO SIGNAL PROCESSING APPLYING THE PROGRAMMING LANGUAGE MATLAB

Namira Krupić,¹ Edin Mujčić²,
^{1,2} Univerzitet u Bihaću, Tehnički fakultet Bihać,
dr. Irfana Ljubijankića bb., namira_krupic@hotmail.com, edin.mujcic@gmail.com

SAŽETAK:

U stvarnom svijetu postoji mnogo fizičkih definicija zvuka, dok u digitalnom svijetu je samo to "hrpa" brojeva raspoređenih u nekim matricnim zapisima. Naravno, sa svakim matricnim zapisom se može upravljati pa tako i sa ovim. Veoma dobro rješenje za obradu matricnih zapisa je programski jezik MATLAB. On omogućava unos, spajanje, razdvajanje, promjenu bilo kog elementa matrice i sve matematičke operacije primijenjene na matrice. Za obradu zvučnih signala u MATLAB-u koristi se „audio and video support“ koji se sastoji od fajlova za: učitavanje i zapisivanje zvučnih signala; fajlova za snimanje i reprodukciju zvučnih signala. Primjenom navedenog „audio and video support-a“ i alata za obradu matrica, u ovom radu, je opisana obrada zvuka od unosa, miksiranja i razdvajanja zvučnih signala, filtriranja zvučnih signala na osnovu frekvencije, proizvodnje „čistih tonova“, grafičkog predstavljanja i reprodukcije zvuka.

Ključne riječi: Zvuk, MATLAB, reprodukcija, snimanje, filter.

ABSTRACT

In the real world there are many physical definition audio, while in the digital world is just a "bunch" of numbers arranged in some matrix write down. Of course, with each matrix write down can be controlled and so with this. Very good solution for the processing of matrix write down the programming language MATLAB. It allows input, mixing, selection, change any element of the matrix and all mathematical operations applied to the matrix. For audio signal processing in MATLAB used is "audio and video support," which consists of files for: read and write audio signals; files for recording and playback of audio signals. Application of aforementioned "audio and video support-a" and tools for processing matrix, in this paper, is described audio processing: from entry, mixing and selection of audio signals, audio signals filtering based on frequency, the production of "pure tone", a graphic representation and audio reproduction.

Keywords: Audio, MATLAB, Reproduction, Recording, Filter.

1. UVOD

Zvuk je fizička pojava koja predstavlja sastavni dio čovjekovog okruženja, gdje se javlja kao prateći element mnogih životnih okolnosti. Zvuk je prisutan gotovo svugdje: u samom organizmu čovjeka, gdje se javlja govor, preko najrazličitih zvukova stalno prisutnih u neposrednom životnom okruženju, pa do zvukova u dubinama okeana ili koji u vidu seizmičkih talasa dopiru iz dubine zemlje [1] i [2]. Sa takvom disperzijom pojavnih oblika zvuk je fizička pojava kojom se danas bave inženjeri raznih oblasti. Oni ga posmatraju i analiziraju sa različitih aspekata i primjenjuju ga u veoma različitim okolnostima. Zvuk je osnova ljudske komunikacije, pa u raznim oblicima zvukova ljudi nalaze određeni smisao i značenja (govor, razni zvučni signali i slično) [3]. Zvuk se zapisuje u dva osnovna formata. Jedan zapisuje zvuk u formi koja odgovara originalnom zvučnom signalu (talasu) i zove se analogno snimanje. Noviji format zapisa je digitalni zapis koji predstavlja zvučni talas kao niz brojeva u binarnom zapisu. Zvuk je u analognom svijetu kontinualan slijed iskazan u vremenu, te ima određeni raspon koji je moguće približno tačno izmjeriti u bilo kojoj vremenskoj tački. Obrada ili procesuiranje zvuka je namjerna promjena audio signala ili zvuka upotrebom audio efekata. Audio signali se predstavljaju u digitalnim ili analognim formatima, pa se i obrada signala može obavljati u analognom ili digitalnom obliku. Obrada signala u analognom obliku se vrši direktno na električnom signalu, dok obrada signala u digitalnom obliku matematički djeluju na digitalnu prezentaciju signala [4]. S toga možemo reći da je digitalna obrada signala DSP (*engl. Digital Signal Processing*) zvuka matematička manipulacija nad signalom kako bi se on na neki način modifikovao ili poboljšao. DSP radi sa diskretnim vremenom i frekvencijama koji su predstavljeni sa nizom brojeva ili simbola. Glavni cilj digitalne obrade signala je mjerenje, filtriranje ili kompresija kontinuiranih stvarnih analognih signala poput zvuka. Obrada počinje pretvaranjem signala iz analognog u digitalni oblik postupkom uzorkovanja te naknadnom digitalizacijom korištenjem konvertora analognih u digitalne signale A/D (*engl. Analog-to-Digital Converter*) [5]. Tako se signal pretvara u niz brojeva ili uzoraka.

U ovom radu je opisana digitalna obrada zvuka upotrebom programskog jezika MATLAB. Pored unosa, generisanja, snimanja i reprodukcije zvuka opisane su i neke specifične korisne operacije za dodatnu obradu zvuka.

1. OSNOVNI POJMOVI O ZVUKU

Zvuk nastaje kada neka materija osciluje, a postoje tri vrste zvuka: prasak, šum i ton. *Prasak* nastaje pri jednom snažnom potresu sa nekoliko brzih treptaja djelića zvučnog izvora, kao što se dešava kod eksplozija. *Šum* je zvuk koji nastaje nepravilnim titranjem zvučnog izvora pri čemu se frekvencija stalno mijenja kao i amplituda. *Ton* nastaje pravilnim titranjem zvučnog izvora i frekvencija je stalna. Ton je jednostavno sinusno titranje. Ton ima određenu frekvenciju. Pored osnovne frekvencije, ton često sadrži i tzv. više harmonike. Osnovnu frekvenciju uho osjeća kao *visinu tona* [6]. Ukoliko je osnovna frekvencija viša i ton je viši. Pored visine tona, čulom sluha razlikujemo i *boju tona*. *Jačina tona* zavisi od amplitude i frekvencije i tretira se kao fizička jačina tona. Pošto uho ima različitu osjetljivost na razne frekvencije onda se uvodi i tzv. subjektivna ili fiziološka jačina tona. Jačina zvuka se često izražava u decibelima, iako ju je moguće izraziti i preko snage koju zvuk nosi ($[W]$, $[W/m^2]$, $[W/rad]$), ili kao efektivni ili maksimalni iznos promjene pritiska u odnosu na pritisak neporemećenog sredstva u kojem se zvuk širi $[Pa]$. Da bi zvučni talas izazvao osjećaj zvuka, mora da ima neku minimalnu jačinu koja se zove *prag čujnosti*. Standardni prag čujnosti se uzima za frekvenciju $1 [kHz]$. Kada jačina zvuka raste mi ga čujemo sve jače dok ne dostigne tzv. *granicu bola*. Čovjek osjeća promjenu jačine zvuka u logaritamskoj skali. Zbog toga se uvodi termin subjektivna jačina zvuka (nivo jačine zvuka) [7]. Zvuk možemo podijeliti i prema frekvenciji [8] na:

1. infrazvuk - ispod $16[Hz]$,
2. ultrazvuk - iznad $20[kHz]$,
3. od $20[Hz]$ do $20[kHz]$ - ljudske granice osjećaja zvuka.

Infrazvuk obuhvata talasne frekvencije $0,1$ do 20 [Hz]. Treba napomenuti da u govoru gotovo uopće nema frekvencija u tom području, te su ove frekvencije u području govora nebitne. Ultrazvuk je zvuk sa frekvencijom većih od gornje granice čujnosti za normalno ljudsko uho od $16-20$ [kHz], u ovisnosti od uzrasta. Ljudsko uho reaguje na zvučne nadržaje samo u jednom intervalu frekvencija koje se nazivaju čujne frekvencije. Nominalne granice čujnog opsega frekvencija utvrđene su na 16 [Hz] (donja) i 20.000 [Hz] (gornja).

3. PROGRAMSKA OBRADA ZVUČNOG SIGNALA

Zvučni signal se u MATLAB-u predstavlja kao vektor čiji su elementi uzorci signala. Kao i druge jednodimenzionalne signale, tako je i audio-signale u MATLAB-u najjednostavnije predstaviti kao uređen niz brojeva odnosno vektor [9]. Pošto je potrebno reprodukovati te signale, pored vrijednosti odabira neophodno je imati informaciju o frekvenciji uzorkovanja. Pošto je standardna reprezentacija brojeva u MATLAB-u double format, efekti kvantizacije signala su zanemarljivi, a do prekoračenja opsega dolazi u izuzetno rijetkim slučajevima.

3.1. Snimanje i učitavanje signala u MATLAB-u

Snimanje zvuka u MATLAB-u je omogućeno pomoću dvije funkcije *audiorecorder* i *wavrecord*, pri čemu je ova posljednja vezana za operativni sistem Windows te neće biti razmatrana u nastavku teksta. Snimanje započinje kreiranjem objekta koji će realizovati snimanje pozivom sljedeće komandne linije:

```
rec = audiorecorder(Fs, nBits, nChannels),
```

gdje je:

Fs - frekvencija uzorkovanja izražena u [Hz] default 8000,
nBits - broj bita default 8 i
nChannels - broj kanala default 1.

Vrijednost frekvencije uzorkovanja zavisi od mogućnosti hardvera, a tipične vrijednosti su 8000, 11025, 16000, 22050 i 44100. Broj bita može biti 8 ili 16, a Windows operativni sistem dozvoljava i 24, dok je maksimalan broj kanala 2. Snimanje započinje komandom:

```
record(rec,duration),
```

gdje je: *rec* objekat, a *duration* dužina segmenta koji treba snimiti izražena u sekundama. Drugi parametar je opcionalni i u slučaju da nije naveden, snimanje se zaustavlja pozivom komande *stop(rec)*. Preslušavanje snimljenog segmenta u objektu *rec* realizuje se pomoću komande:

```
play(rec,[start stop]),
```

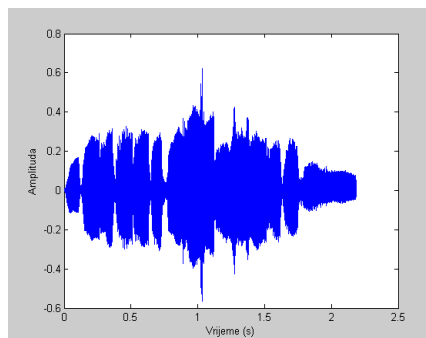
pri čemu su parametri *start* i *stop* opcionalni, odnosno ako nije naveden ni jedan od njih, biće reprodukovan cijeli fajl; ako je naveden samo jedan, onda se podrazumijeva da je u pitanju parametar *start* i reprodukcija započinje od onog odabira čiji je *start* indeks i traje sve do posljednjeg odabira; i naravno, ako su navedena oba parametra, onda se reprodukuje signal od odabira sa indeksom *start* do odabira sa indeksom *stop*. Za učitavanje snimljenog fajla u MATLAB-ov *Workspace* koristi se naredba:

```
smp = getaudiodata(rec,dataType),
```

gdje je: *dataType* format učitanih podataka ('double', 'single', 'int16', 'int8' i 'uint8') default vrijednost je 'double'. Proces snimanja zvuka se dodatno kontroliše komandama *pause(rec)* i *resume(rec)*, kojima se snimanje privremeno zaustavlja i pokreće. Za učitavanje zvukova može se

koristiti i naredba *wavread*. Sa dovoljno visokom frekvencijom uzorkovanja, vektor sa dvostrukom preciznošću ima dovoljnu rezoluciju za bilo koji oblik procesuiranja koji se mora obaviti što znači da se problemi sa kvantizacijom mogu u potpunosti zanemariti [10]. Primjer učitavanja, predstavljanja i reprodukcije zvučnog signala dat je u nastavku ovog rada.

```
[zvuk,fs]=wavread('flute.wav');
vrijeme=(1/44100)*length(zvuk);
t=linspace(0,vrijeme,length(zvuk));
plot(t,zvuk)
xlabel('Vrijeme (s)');
ylabel('Amplituda');
```



Slika 1: Graf zvučnog signala

3.2. Reprodukcija i generisanje zvuka

Za reprodukciju zvuka upotrebljava se naredba *sound*. Pri tome možemo izabrati mono ili stereo, tj. slušati pojedinačno jedan ili drugi kanal ili oba zajedno. Opći oblik poziva ove naredbe je:

$$\text{sound}(\text{sig},Fs,nBits),$$

gdje je: *sig* -vektor ili matrica koji sadrži odbirke zvučnog signala, *Fs* - frekvencija uzorkovanja i *nBits* -broj bita po uzorku (8 ili 16). Rezultat je zvuk koji se prosljeđuje na izlazni audio uređaj. U slučaju da se ne navedu, podrazumijevane vrijednosti za frekvenciju uzorkovanja i broj bita po uzorku su 8000 Hz i 8 bita. Ako se želi postići veća fleksibilnost prilikom reprodukcije, upotrebljava se naredba *audioplayer*. Opći oblik poziva je:

$$\text{ap} = \text{audioplayer}(\text{sig},Fs,nBits),$$

gdje je: *sig*- vektor ili matrica koja sadrži odbirke audio-signal, *Fs* - frekvencija uzorkovanja, *nBits* - broj bita po uzorku, a rezultat je *audioplayer* objekat *ap*. Sama reprodukcija započinje pozivom funkcija *play*, *resume* ili *playblocking*. Opći oblik poziva funkcije *play* je:

$$\text{play}(\text{ap},\text{what}),$$

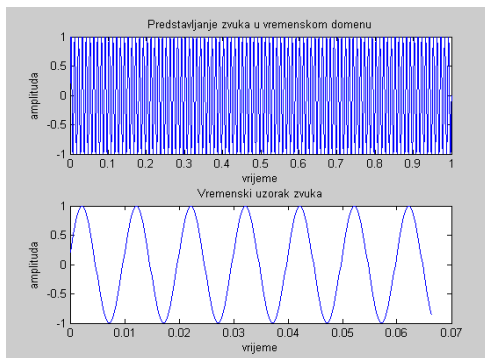
gdje je: *ap* -audioplayer objekat, *what* -opcionalni parametar koji definiše šta će se reprodukovati.U slučaju da se *what* ne navede, podrazumijeva se da reprodukcija počinje od početka datoteke, ako se navede samo jedan broj on se tumači kao indeks odabirka od koga treba započeti reprodukciju, a ako se navedu dva broja, prva vrijednost predstavlja indeks početnog, a druga indeks krajnjeg odabirka u toku reprodukcije. Identičnu sintaksu ima i funkcija *playblocking*. Razlika između ove dvije funkcije je da u slučaju funkcije *playblocking* nije moguće zaustavljanje reprodukcije pomoću komandi *stop* i *pause*. Reprodukcija se može pokrenuti i pomoću funkcije *resume(ap)* pri čemu reprodukcija započinje od prvog odabirka, ako prije toga nijednom nije pozvana funkcija *play(ap)* ili ako je prethodna reprodukcija zaustavljena pomoću funkcije *stop(ap)*, odnosno od odabirka gdje je zaustavljena reprodukcija pomoću funkcije *pause(ap)*. Vizualizacija talasnog oblika audio-signal jednostavno se realizuje pomoću funkcija *plot* i *stem*. Pored talasnog oblika za analizu govora i zvuka interesantn je i spektar signala odnosno njegove promjene u toku vremena, tzv. spektrogram. Spektrogram je realizovan u MATLAB-u u okviru funkcije *spectrogram*. Za istraživanje zvuka je ponekad potrebno i generirati zvukove. Najčešće je potrebno generirati čiste tonove, šum ili varijabilne tonove. Novi zvuk je također moguće dobiti sabiranjem već postojećih zvukova [11] i

[12]. Čisti ton je nemodulirani sinusni oblik. Sinusna funkcija omogućuje da se generiše ton s frekvencijom F_t sa frekvencijom uzorkovanja F_s koji traje T_d sekundi sljedećom naredbom:

$$s = \sin([1:F_s*T_d]*2*\pi*F_t/F_s);$$

Za generisanje tona koji se nalazi na frekvenciji od $100 [Hz]$ u trajanju od $1 [s]$ sa frekvencijom uzorkovanja od 3000 uzoraka/sekundi i predstavljanje generisanog tona može se koristiti M-file:

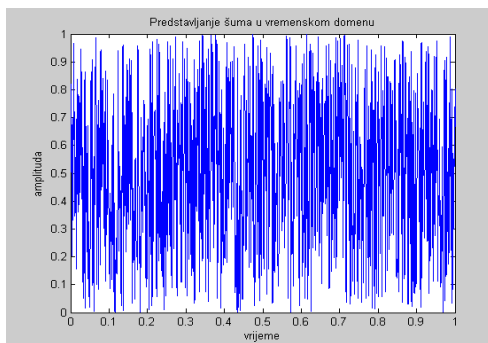
```
s = sin([1:3000*1]*2*pi*100/3000);
t=linspace(0, 1, length(s));
subplot(2,1,1),plot(t, s)
xlabel('vrijeme')
ylabel('amplituda')
title('Predstavljanje zvuka u vremenskom
domenu')
subplot(2,1,2),plot(t(1:200), s(1:200))
xlabel('vrijeme')
ylabel('amplituda')
title('Vremenski uzorak zvuka')
```



Slika 2: Predstavljanje zvuka u MATLAB-u

Šum se obično sastoji od ravnomjerno raspoređenih slučajnih brojeva koje je u MATLAB-u jednostavno generisati korištenjem funkcije *rand*. Za generisanje šuma u trajanju T_d sekundi sa frekvencijom uzorkovanja F_s može se koristiti M-file:

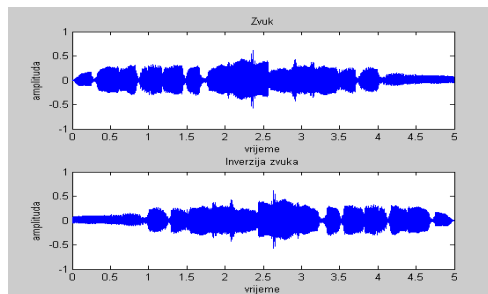
```
Fs=1000;
Td=1;
s = rand(1, Fs*Td);
t=linspace(0, 1, length(s));
plot(t,s)
xlabel('vrijeme')
ylabel('amplituda')
title('Predstavljanje šuma u vremenskom
domenu')
```



Slika 3: Predstavljanje šuma u MATLAB-u

Za reprodukciju zvuka unazad je dovoljno obrnuti uzorke u vektoru zvuka tako da se podaci koji su bili na početku nađu na kraju, a oni s kraja na početku. Za to je moguće iskoristiti funkciju *flipud*.

```
[zvuk,fs]=wavread('flute.wav');
zvuk_unazad=flipud(1_r_zvuk); soundsc(zvuk,fs);
pause(1)
soundsc(zvuk_unazad, fs);
t=linspace(0, 5, length(zvuk));
subplot(2,1,1), plot(t,zvuk)
xlabel('vrijeme')
ylabel('amplituda')
title('Zvuk')subplot(2,1,2), plot(t,zvuk_unazad)
xlabel('vrijeme')
ylabel('amplituda')
title('Inverzija zvuka')
```



Slika 4: Inverzija zvučnog signala

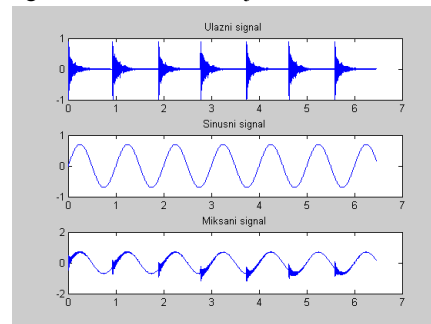
3.3. Kombinovanje zvučnih signala

Kombinovanje signala nastaju novi zvukovi. Osnovni oblik kombinovanja je spajanje zvučnih signala sejski [11]. U MATLAB-u se ova operacija svodi na nastavljanje vektora pomoću konkatencije. Na primjer:

```
[r_zvuk,fs]=wavread('road.wav');  
zvuk_nastav = [r_zvuk; r_zvuk; r_zvuk];  
soundsc(zvuk_nastav, fs);
```

Miksanje signala predstavlja miješanje dva ili više zvučnih signala u određenom omjeru u jedan signal. Poželjno je da dobiveni signal bude privlačan za slušaoca. MATLAB omogućava miksanje zvučnih signala s tim da frekvencije uzorkovanje i trajanje zvučnih signala budu identični [12]. Mi zahvaljujući velikim mogućnostima MATLAB-a uvijek možemo zadovoljiti tražene uslove. U nastavku ovog rada je naveden program za miksanje dva signala u određenom omjeru.

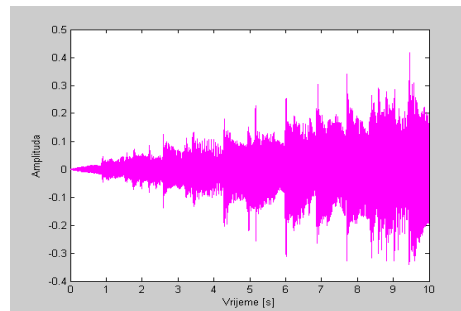
```
[y, Fs, nbits] = wavread('toms.wav');  
t=0:1/Fs:1/Fs*(length(y)-1);  
sinusnisignal = 0.7*sin(2*pi*t);  
novisignal =0.5* y + sinusnisignal;  
subplot(3,1,1), plot(t,y), title('Ulazni signal ');  
subplot(3,1,2), plot(t,sinusnisignal), title('Sinusni  
signal');  
subplot(3,1,3), plot(t,novisignal), title('Miksani  
signal');  
wavplay(novisignal)
```



Slika 5: Miksanje dva zvučna signala

Pored običnog množenja skalarom, signal je moguće množiti i sa drugim signalima. Za to se najčešće koriste urađene MATLAB funkcije za generiranje signala poput *step*, *ramp*, *sawtooth*, *square*, *chirp* i niz drugih. Bilo koji vektor vrijednosti koji ima istu dužinu kao zvuk se može koristiti za pojačanje ili prigušenje signala. To je navedeno u sljedećem M-fajlu:

```
[gitara, fs] = wavread('guitar.wav');  
ramp=(0:1/(44100*10):1);  
izl_signal = gitara(1:length(ramp)).*ramp;  
t=linspace(0,10, length(ramp));  
plot(t, gitara(1:length(ramp)).*ramp);  
xlabel('Vrijeme [s]');  
ylabel('Amplituda');
```



Slika 6: Linearno pojačanje zvučnog signala

3.4. Digitalni filteri

Filteri vrše filtriranje zvučnog signala po frekvenciji. Njihov zadatak je da određene frekvencije neutrališu (priguše), a određene propuste bez ikakve promjene [13]. Postoji više vrsta filtera[14][15], a to su:

- niskopropusni filteri (engl. *LowPass*),
- visokopropusni (engl. *HighPass*),
- pojasnopropusni (engl. *BandPass*).

U MATLAB-u postoje naredbe za projektovanje navedenih filtera. Za niskopropusni filter tipa Chebyshev II je:

$$[b,a] = \text{cheby2}(N,R,Wn,'low')$$

pri čemu je:

N - red filtra,

R - prigušenje u $[db]$,

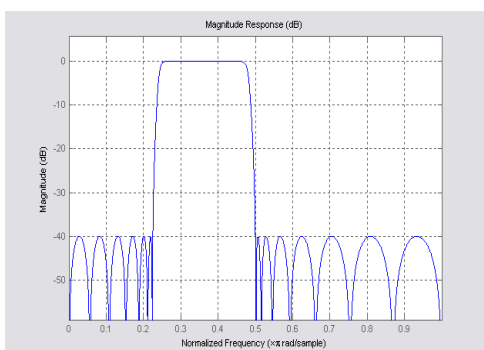
Wn - granična frekvencija područja prigušenja normirana sa polovinom frekvencije uzorkovanja.

Za visokopropusni filter naredba je:

$$[b,a] = \text{cheby2}(N,R,Wn,'high')$$

Kao rezultat se dobivaju koeficijenti polinoma brojnika i nazivnika IIR filtra. U nastavku je projektovan uskopojasni filter kombinacijom niskopropusnog i visokopropusnog filtera.

```
fs=8000;
fd=[2*1000/fs];
fn=[2*900/fs];
R=40;
[N,Wn]=cheb2ord(fd,fn,1,R)
[bh,ah]=cheby2(N,R,fn,'high');
fd=[2*2100/fs];
fn=[2*2000/fs];
[N,Wn]=cheb2ord(fd,fn,1,R)
[bl,al]=cheby2(N,R,fn,'low');
b=conv(bh,bl); % Koeficijenti brojnika
kaskade
a=conv(ah,al); % Koeficijenti
nazivnika kaskade
fvtool(b,a)
```



Slika 7: Projektovani filter

pri čemu su: bh i bl brojnici, a ah i al nazivnici VP i NP filtra. Svaki učitani signal će biti filtriran primjenom upravo projektiranog pojasnopropusnog filtra. U tu svrhu koristi se MATLAB-ova funkcija za filtraciju:

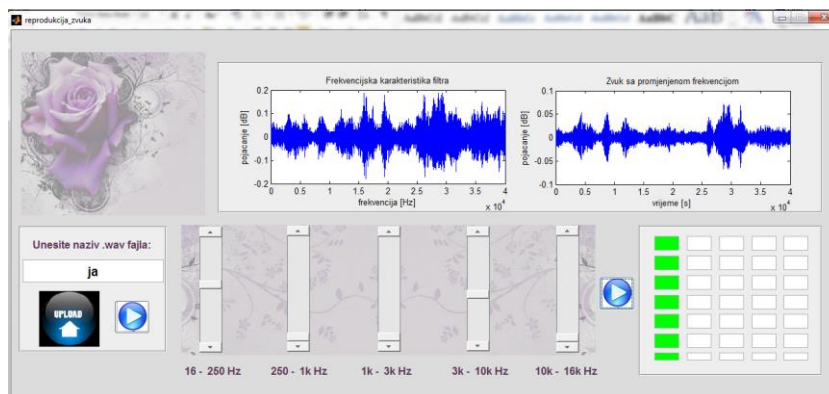
$$yf = \text{filter}(b,a,y);$$

pri čemu su: b i a - koeficijenti filtra, a y - originalni nefiltrirani zvučni signal.

3.5. Obrada zvuka upotrebom grafičkog sučelja

Upotrebom grafičkog korisničkog sučelja (engl. *Graphical user interface*) u MATLAB-u možemo kreirati grafičko sučelje. Primjenom ranije opisanih funkcija za obradu zvuka i grafičkih objekata koji su na raspolaganju možemo napraviti grafičko sučelje za obradu zvučnih signala (vidi sliku 8).

Na ovaj način je omogućeno korisniku da unese snimljeni zvučni signal i klikom na strelicu, ispod prostora za unos snimljenog zvučnog signala, pokrene reprodukciju i prikaz zvučnog signala. Upotrebom ekvilajzer-a možemo prilagoditi zvuk svojim zahtjevima i klikom na strelicu na desnoj strani grafičkog sučelja vrši se reprodukcija podešenog signala i prikaz prilagođenog zvučnog signala.



Slika 8: Obrada zvuka upotrebom grafičkog sučelja

Sa slike 8 možemo zaključiti da je obrada zvuka u MATLAB-u upotrebom grafičkog sučelja vrlo jednostavna i pregledna.

4. ZAKLJUČAK

Na osnovu navedenih programa može se zaključiti da MATLAB posjeduje veoma dobre alate za obradu zvuka. Zvučni signali se posmatraju kao vektori ili matrice (stereo signali) a MATLAB u osnovi radi sa poljima (tj. u slučaju brojeva matricama). Tako da je analiza zvučnih signala veoma prilagodljiva načinu obrade informacija u MATLAB-u. Također, upotrebom grafičkog sučelja i jednostavnog formiranja EXE fajla sve obrade postaju dostupne gotovo svim uzrastima.

5. LITERATURA

- [1] <http://hr.wikipedia.org/wiki/Zvuk>(12.11.2014)
- [2] https://www.fer.unizg.hr/_download/repository/ZIO_01_Osnove_zvuka_2014.pdf (18.11.2014)
- [3] I. Gazdić, *Fizika (Odabrana poglavlja za tehničke fakultete)*, Tuzla 2009.
- [4] E. Jakupović, M.Kerenović, *Fizika III (Univerzitetska knjiga Bihać)*,1998.
- [5] Douglas F. Elliott, *Digital Signal Processing*, Rockwell International Corporation Anaheim, California 1987.
- [6] <http://981n.blogspot.com/2013/11/ton-boja-i-visna-ona-izvori-zvuka.html> (11.12.2014)
- [7] <http://physics.blogger.ba/arhiva/2010/02/04/2426249> (11.12.2014)
- [8] http://en.wikipedia.org/wiki/Frequency_spectrum (16.12.2014)
- [9] I. McLoughlin, *Applied Speech and Audio Processing: With MATLAB Examples*, 2009.
- [10] M.Popović i A.Mojsilović, *Digitalna obrada signala*, Nauka, 1996.
- [11] M. Kahrs, K. Brandenburg, *Application of Digital Signal Processing to Audio and Acoustics*, Kluwer Academic Publishers, 2002
- [12] L. R. Rabiner, J. H. McClellan, T.W. Parks, *FIR Digital Filter Design Techniques Using Weighted Chebyshev Approximation, Proc IEEE, Vol. 63, pp 595-610*
- [13] http://dog.zesoi.fer.hr/laboratorij/pdf/Skripta_za_Labose_iz_DOGa.pdf(15.12.2014)
- [14] http://www.mathworks.com/tagteam/55876_digfilt.pdf (22.12.2014)
- [15] K. V. Madisetti, *Digital Signal Processing Fundamentals*, CRC Press, 2010.