

Mikroelektronski sistemi

Laboratorijske vaje

Cikel 1: Uvod v MATLAB/SIMULINK

Namen uvodnih je spoznati se z:

- Orodno vrstico v okolju MATLAB
- Pripravo datotek .m
- Okoljem SIMULINK

Vaja1:

- a) Pregled osnovnih funkcij v okolju MATLAB;
- b) Priprava časovnega vektorja;
- c) Uporaba elementarnih funkcij;
- d) Uporaba fft algoritma;
- e) Prikaz rezultatov v časovnem in frekvenčnem prostoru;

Pomoč:

- help elfun - ukaz ki vrne spisec elementarnih funkcij;
- linspace -funkcija ki generira vektor v katerem števila linearno naraščajo;
- figure() - ukaz ki generira grafični objekt na zaslonu v katerem matlab prikaže zahtevane spremenljivke;
- plot - ukaz ki izriše funkcijo ene spremenljivke
- hold - funkcija, ki omogoča izris večjega števila grafov v istem grafičnem objektu
- fft – Fast Fourier transform

Navodila: Pripravite .m datoteko v kateri ustvarite časovni vektor. Časovni vektor uporabite pri prikazu dveh trigonometričnih funkcij na skupnem grafu.

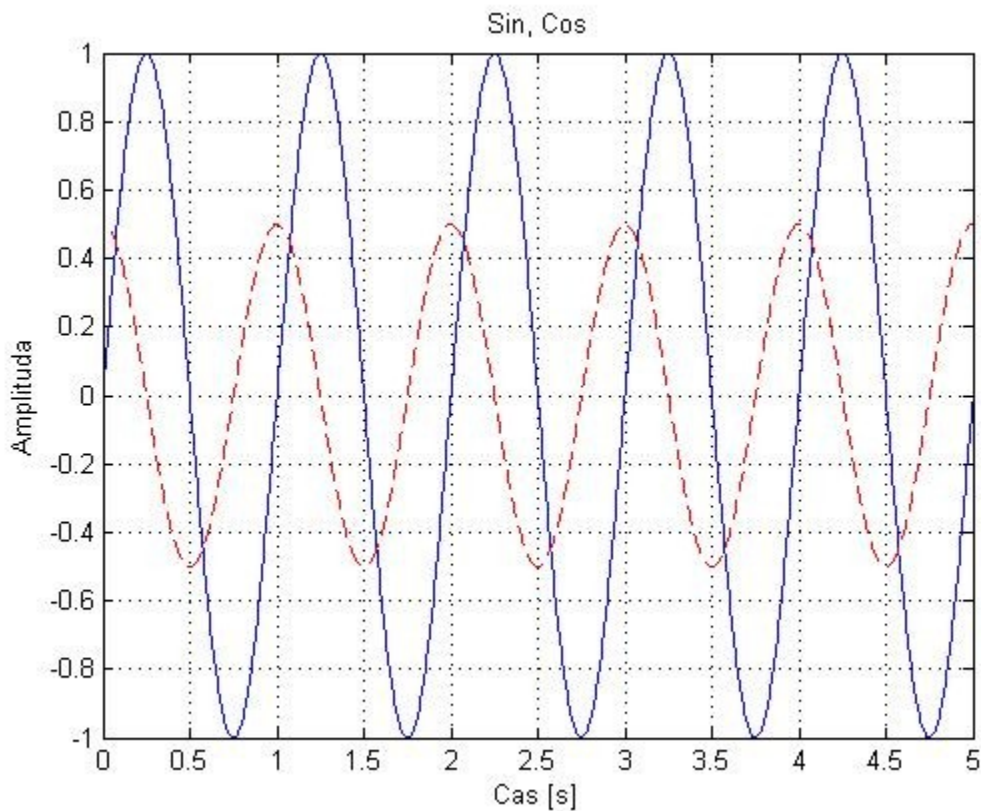
Primer:

```
clear all
close all
help effun

t=linspace(0,5,1000); % priprava časovnega vektorja s časovnim korakom 0.5ms
%%% Signala za izris %%%
y=sin(2*pi.*t);
y1=0.5.*cos(2*pi.*t);

%%% izris rezultatov %%%
figure()
plot(t,y)
hold
plot(t,y1,'--r')
grid

title('Sin, Cos');
xlabel('Cas [s]');
ylabel('Amplituda');
```



Slika 1: Primer uporabe elementarnih funkcij in časovnega prikaza

Vaja 2:

- a) Uporaba fft funkcije;
- b) Prikaz v časovnem in frekvenčnem prostoru

Navodila: Pripravite .m datoteko v kateri seštejete dve trigonometrični (sinusni) funkciji različnih frekvenc in amplitud. Vsoto in trigonometrični funkciji izrišite na skupnih grafih v časovnem in frekvenčnem prostoru. Za analizo v frekvenčnem prostoru uporabite fft() funkcijo. Informacije o fft funkciji dobite z ukazom "help fft"

Primer:

$$x_1 = A_1 \sin(2\pi f_1 \cdot t)$$
$$x_2 = A_2 \sin(2\pi f_2 \cdot t)$$
$$y = x_1 + x_2$$

```
clear all
close all
f1=60; % frekvenca prvega signala
f2=180; % frekvenca drugega signala

A1=2; % amplituda prvega signal
A2=1; % amplituda drugega signal

N=2^14; % število časovnih korakov
Fs=2^14;
Ts=1/Fs; % časovni korak
t=(1:N-1).*Ts; % časovni vektor

%%%% časovna signala
x1=A1.*sin(2*pi.*t*f1);
x2=A2.*sin(2*pi.*t*f2);

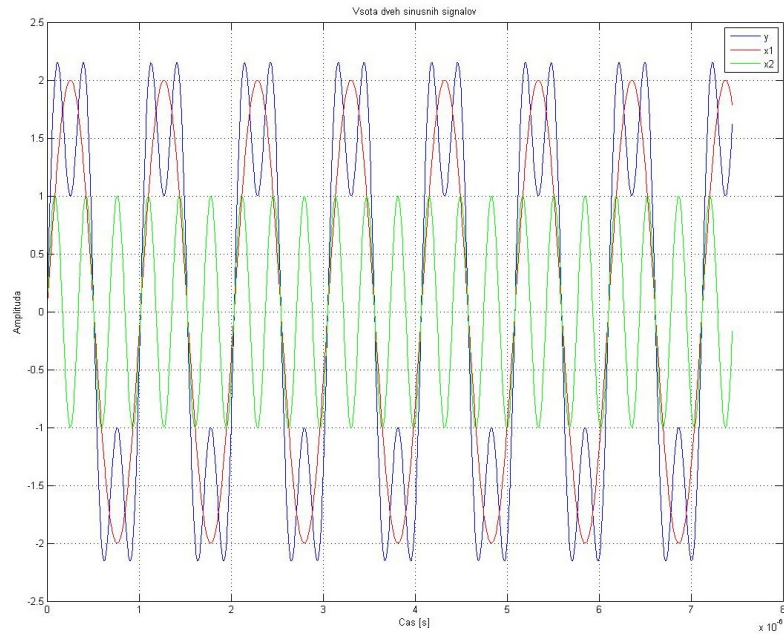
y=x1+x2; %%%%% vsota signalov

%%%% Izris %%%
figure();
plot(t(1:2000).*1/N,y(1:2000))
hold
plot(t(1:2000).*1/N,x1(1:2000),'r')
plot(t(1:2000).*1/N,x2(1:2000),'g')
grid;
hleg1=legend('y','x1','x2');

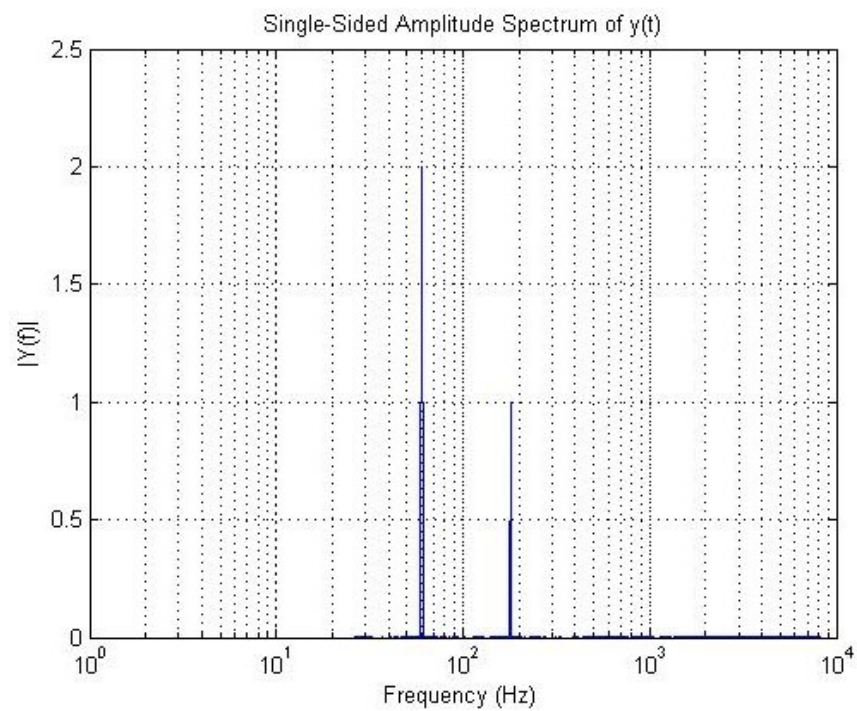
%%%% uporaba funkcije fft %%%

L=length(t);
NFFT = 2^nextpow2(L); % Next power of 2 from length of y; pospeši fft
Y = fft(y,NFFT)/L; % izračun fft-ja
f = Fs/2*linspace(0,1,NFFT/2+1); % priprava frekvenčnega vektorja
```

```
% Plot single-sided amplitude spectrum.
figure()
semilogx(f,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
grid;
```



Slika 2: Primer vsote dve sinusnih signalov različnih frekvenc



Slika 3: FFT vsote

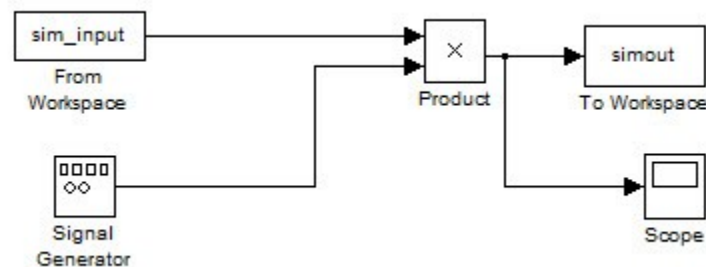
Vaja 3:

- a) Uporaba fft funkcije;
- b) Deklaracija funkcije
- c) Simulink

Navodila: Pripravite .m datoteko v kateri množite dve trigonometrični (sinusni) funkciji različnih frekvenc in amplitud. Rezultat množenja prikažite v frekvenčnem prostoru. Množenje signalov naj poteka v okolju simulink, ki ga zaženete z ukazom simulink v orodni vrstici.

Primer:

$$A_1 \sin(\omega_1 t) \cdot A_2 \sin(\omega_2 t) = \frac{1}{2} (A_1 A_2 \cos(\omega_1 t - \omega_2 t) - A_1 A_2 \cos(\omega_1 t + \omega_2 t))$$



Slika 4: Realizacija množenja v okolju simulink

Deklaracija fft_izrisa:

```
function []=fft_izris(y,Fs)
% function []=fft_izris(y,Fs)
% y = vhodni podatek;
% Fs= vzor?evalna frekvenca

L=length(y);
w=hanning(L);
y=y.*w.*2;

NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
figure()
semilogx(f,2*abs(Y(1:NFFT/2+1)))
ax=axis();
axis([ax(1) 10e3 ax(3) ax(4)])
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
grid;
```

```

clear all;
close all;

A=1;      %    amplituda signala
f1=100;   %    frekvenca prvega signala
f2=150;   %    frekvenca drugega signala
Fs=2000;  %    vzor?evalna frekvenca

N=2^12;   %    število ?asovnih korakov
Ts=1/Fs;
t=(1:N-1).*Ts; % časovni vektor

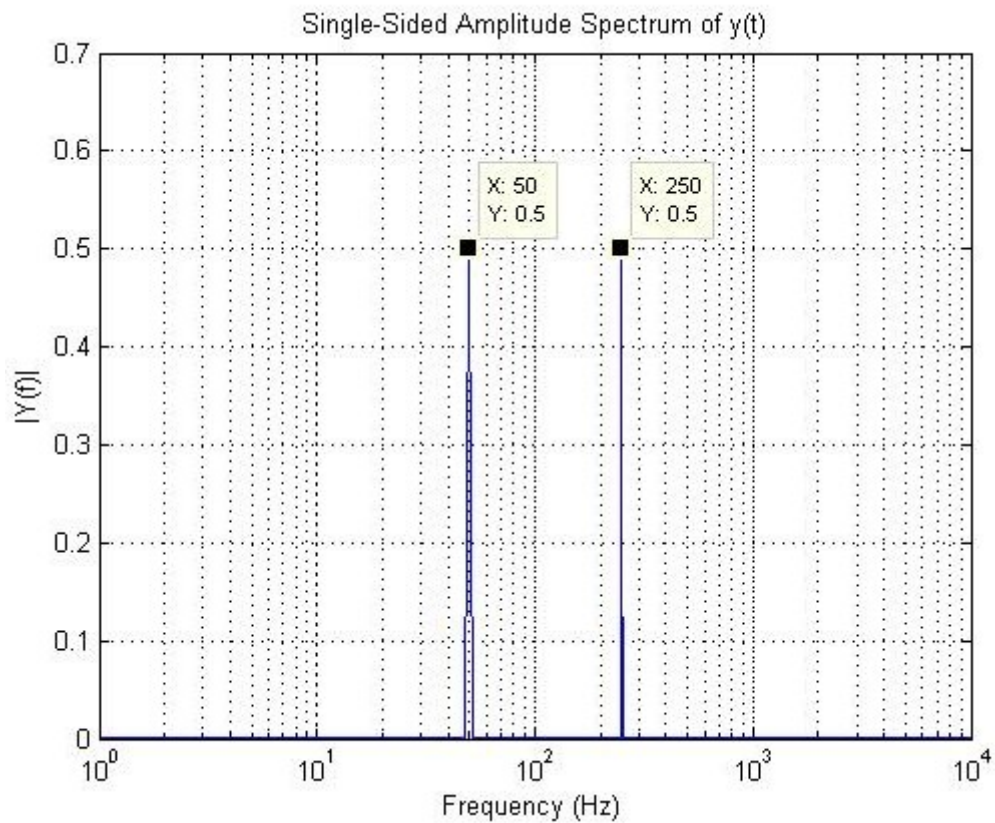
noise=0.005*randn(size(t)); % dodan šum
x=A*cos(2*pi.*t*f1)+noise; % vhodni signal + šum
sim_input=[t' x'];          % priprava vhodne spremenljivke za simulink

options=simset('RelTol', 1e-3, 'FixedStep', 1/Fs, 'MaxStep', 1/Fs); % parametri
simulink simulacije

sim('cikell_v3', (N-1)*Ts, options); % zagon modela v simulink okolju help sim

fft_izris(simout, Fs)

```

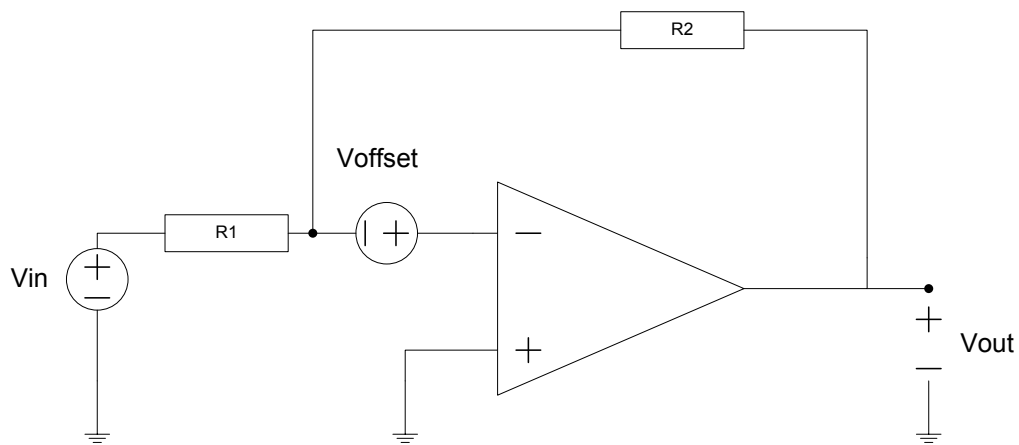


Slika 5: Rezultat množenja dve trigonometričnih funkcij

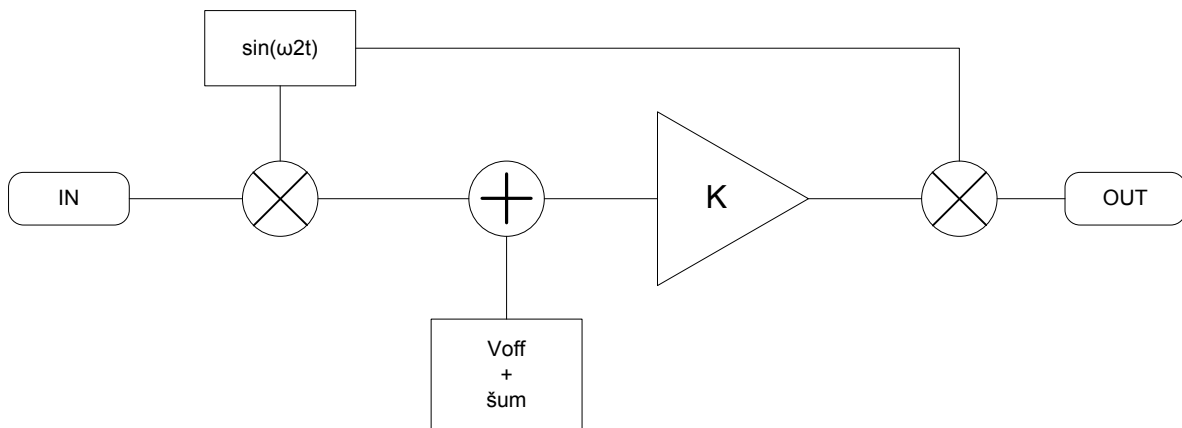
Vaja 4:

- Priprava funkcije izrisa
- Uporaba simulink okolja za preprost model ojačevalnika z dodano ničelno napetostjo
- Časovna in frekvenčna analiza "chopping" tehnike

Eden od načinov izničenja nezaželene ničelne napetosti ojačevalnikov je "chopping" tehnika. V okolju simulink modelirajte vezje na Slika 6 ter uporabite "chopping" tehniko za izničenje nezaželene ničelne napetosti. Rezultat prikažite v frekvenčnem prostoru.



Slika 6: Primer ojačevalnika

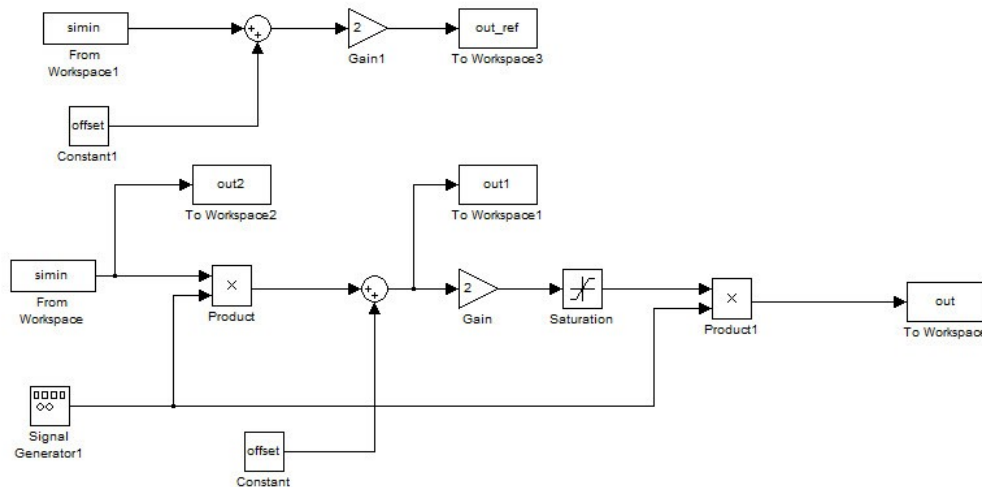


Slika 7: Ojačevalnik in "chopping" tehnika

Primer: Slika 76 prikazuje ojačevalnik, ki vsebuje ničelno napetost in šum pri nizkih napetostih. Model ojačevalnika, ki ga realiziramo v okolju simulink prikazuje Slika 7. Izhodno napetost izračunamo kot :

$$V_{in} = A_1 \sin(\omega_1 t)$$

$$V_{out} = V_{in} \cdot \sin(\omega_2 t) \cdot K \cdot \sin(\omega_2 t) = \frac{1}{4} (2K \sin(\omega_1 t) - A_1 K \sin(\omega_1 t - 2\omega_2 t) - A_1 K \sin(\omega_1 t + 2\omega_2 t))$$



Slika 8: Model ojačevalnika v simulink okolju

```
close all
clear all

Fs=2^18;      % vzorčevalna frekvenca
fmux=2^10;    % frekvenca chopping signala

N=2^16; % število časovnih korakov
Ts=1/Fs;
t=(1:N-1).*Ts; % časovni vektor

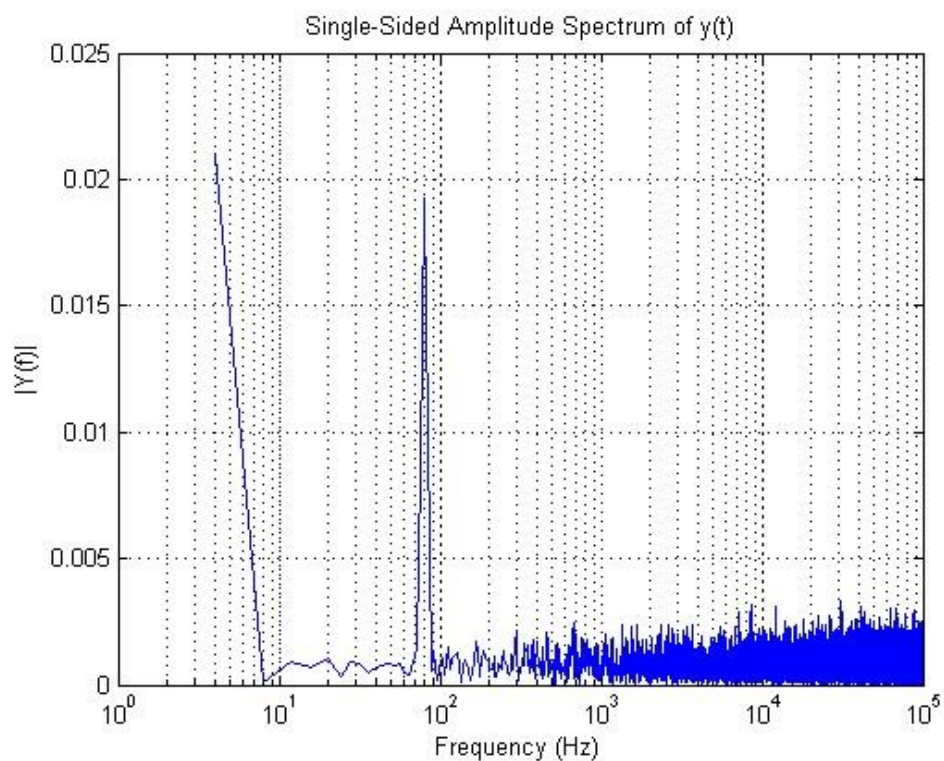
noise=0.05*randn(size(t)); % šum

A1=0.01;      % amplituda vhodnega signala
f=80;         % frekvenca vhodnega signala

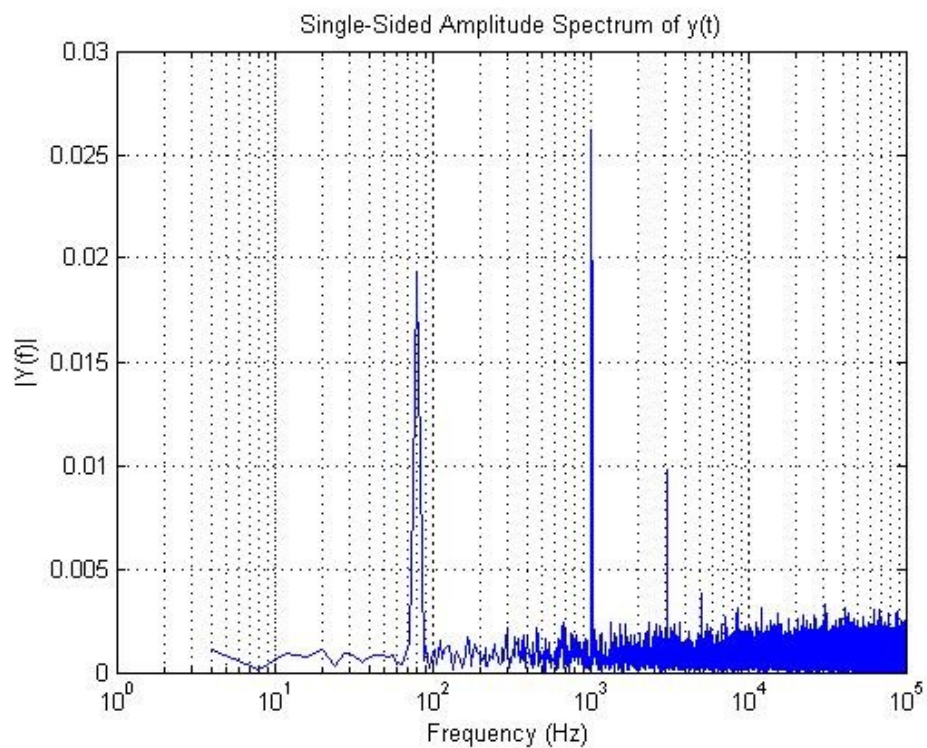
x=A1*sin(2*pi.*t*f)+noise; % vhodni signal
simin=[t' x'];
offset=1e-2;   % ničelna napetost

options=simset('RelTol', 1e-3, 'MaxStep', 1/Fs, 'FixedStep', 1/Ts);
sim('cikell_v4a', (N-1)*Ts, options); % zagon simulink modela

% Prikaz rezultatov v frekvenčnem prostoru
fft_izris(out, Fs);
fft_izris(out_ref, Fs);
```

Slika 9: Spekter izhodne napetosti, kjer je prisotna ničelna napetost



Slika 10: Spekter izhodne napetosti s "chopping" tehniko brez prisotnosti ničelne napetosti