

## DSZOB, cvičenie 2.

### Zadanie:

#### Úloha 1 – Základné generovanie signálov

Vygenerujte a vhodne vizualizujte nasledovné signály (vzorkovacia frekvencia 44.1 kHz, časový úsek 3 sekundy):

1. Sínusový signál s frekvenciou zvoleného tónu temperovaného ladenia (vid tab. dole) a magnitúdou = 0,6.
2. Sínusový signál s frekvenciou iného zvoleného tónu temperovaného ladenia (vid tab. dole) a magnitúdou z rozsahu  $<0,2:0,5>$ .
3. Sínusový signál s frekvenciou iného zvoleného tónu temperovaného ladenia (vid tab. dole) a magnitúdou z rozsahu  $<0,1:0,4>$ .
4. Generujte zložený signál ako súčet z predchádzajúcich vygenerovaných signálov.
5. K zloženému signálu aditívne pripočítajte šum s magnitúdou z rozsahu  $<0,01:0,05>$ .

(pomôcka: funkcia *rand()* alebo *randn()* )

Vygenerované signály si vypočítajte po sebe v poradí generovania (zložený signál na záver).

Vygenerované signály vizualizujte!

Pozn.: Môžete sa pokúsiť o akord (vid tab. dole)

#### Tab. Temperované ladenie - frekvencie

	0	1	2	3	4	5	6	7	8	9
c	16,35	32,7	65,4	130,8	261,6	523,2	1046,4	2092,8	4185,6	8371,2
cis	17,32	34,64	69,29	138,58	277,16	554,31	1108,62	2217,24	4434,49	8868,98
d	18,35	36,7	73,41	146,82	293,64	587,27	1174,54	2349,09	4698,18	9396,35
dis	19,44	38,89	77,77	155,55	311,1	622,19	1244,39	2488,77	4977,55	9955,09
e	20,6	41,2	82,4	164,8	329,6	659,19	1318,38	2636,76	5273,53	10547,05
f	21,82	43,65	87,3	174,6	349,19	698,39	1396,78	2793,55	5587,11	11174,21
fis	23,12	46,24	92,49	184,98	369,96	739,92	1479,83	2959,67	5919,33	11838,66
g	24,5	48,99	97,99	195,98	391,96	783,91	1567,83	3135,66	6271,31	12542,63
gis	25,95	51,91	103,82	207,63	415,26	830,53	1661,06	3322,11	6644,23	13288,45
a	27,5	54,99	109,99	219,98	440,00	879,91	1759,83	3519,66	7039,31	14078,62
ais	29,13	58,26	116,53	233,06	466,12	932,24	1864,47	3728,95	7457,89	14915,78
h	30,86	61,73	123,46	246,92	493,84	987,67	1975,34	3950,68	7901,36	15802,72

Príklady akordov / Tóny, z ktorých sa skladá:

- Cdur / C, E, G
- Gdur / G, B, D
- Amoll / A, C, E

Postup vhodne dokumentuje (Code/Text bloky)!

## Riešenie:

```
% Riesenie / Solution
```

```
f_ais5 = 932.24
```

```
%vybraný ton č.1
```

```
f_ais5 = 932.2400
```

```
f_0 = 44.1  
všetky tóny 1.Úlohy
```

```
%vzorkovacia frekvencia pre
```

```
f_0 = 44.1000
```

```
mag_ais5 = 0.6
```

```
%1. amplitúda
```

```
mag_ais5 = 0.6000
```

```
x_0 = 0:1/(f_0 * 1000):3  
sampoalo
```

```
%vector "časov" kedy sa
```

```
x_0 = 1x132301  
0 0.0000 0.0000 0.0001 0.0001 0.0001 0.0001 0.0002 ...
```

```
%1. Sínusový signál s frekvenciou ais 5 a magnitúdou = 0,6:
```

```
sinY_ais5 = sin(2*pi*x_0*f_ais5) * mag_ais5  
pôvodného ais 5 tónu.,
```

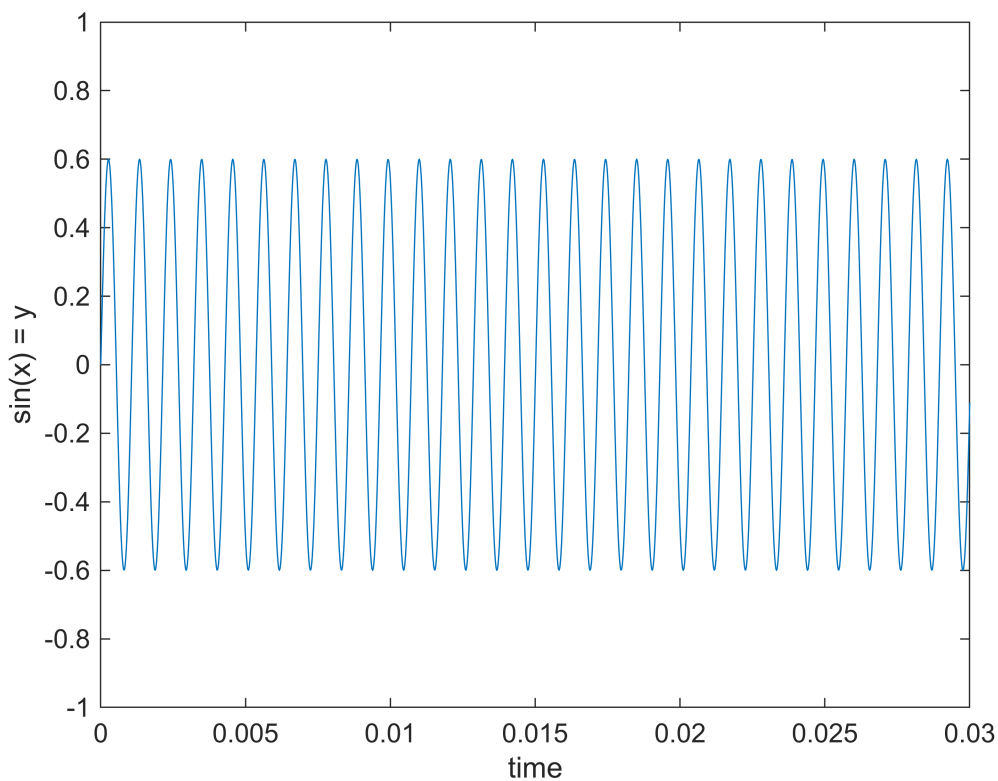
```
%zdigitalizovaný signál z
```

```
sinY_ais5 = 1x132301  
0 0.0795 0.1575 0.2328 0.3040 0.3698 0.4291 0.4809 ...
```

```
plot(x_0,sinY_ais5)  
xlabel("time")  
ylabel("sin(x) = y")  
xlim([0 0.03])  
hranice x osi  
ylim([-1 1])  
hranice y osi
```

```
%nastavenie grafu, vzhladom na
```

```
%nastavenie grafu, vzhladom na
```



%2. Sínusový signál s frekvenciou iného zvoleného tónu temperovaného ladenia (vid tab. dole) a magnitúdou z rozsahu <0,2:0,5>.

```
f_fis4 = 369.96
```

%vybraný ton č.2

```
f_fis4 = 369.9600
```

```
mag_fis4 = mod(rand() * 2, 0.5)
amplitúdy, z intervalu 0.2-0.5
```

%náhodný výber hodnoty

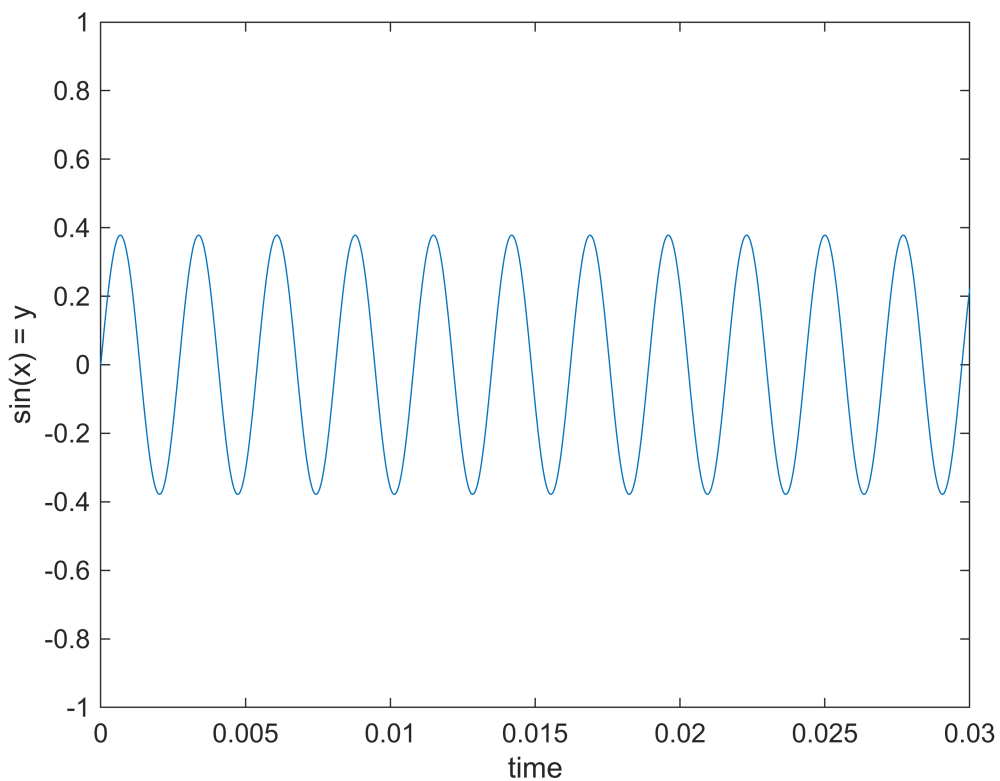
```
mag_fis4 = 0.3782
```

```
sinY_fis4 = sin(2 * pi * x_0 * f_fis4) * mag_fis4
pôvodného fis 4 tónu.
```

%zdigitalizování signál z

```
sinY_fis4 = 1×132301
           0      0.0199      0.0398      0.0595      0.0791      0.0985      0.1176      0.1364 ...
```

```
plot(x_0, sinY_fis4)
xlabel("time")
ylabel("sin(x) = y")
xlim([0 0.03])
ylim([-1 1])
```



%3.Sínusový signál s frekvenciou iného zvoleného tónu temperovaného ladenia (vid tab. dole) a magnitúdou z rozsahu <0,1:0,4>.

f\_d2 = 73.41

%vybraný ton č.3

f\_d2 = 73.4100

mag\_d2 = mod(rand() + 0.1, 0.4)  
amplitúdy, z intervalu 0.1-0.4

%náhodný výber hodnoty

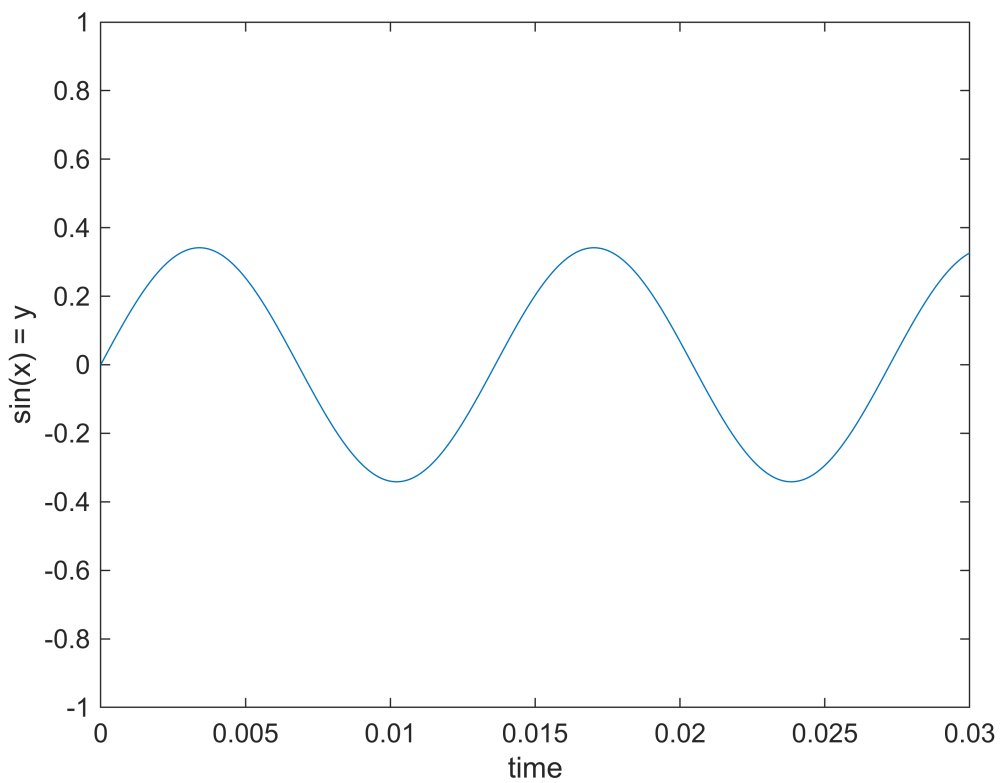
mag\_d2 = 0.3415

sinY\_d2 = sin(2 \* pi \* x\_0 \* f\_d2) \* mag\_d2  
pôvodného d 2 tónu.

%zdigitalizování signál z

```
sinY_d2 = 1×132301
          0      0.0036    0.0071    0.0107    0.0143    0.0179    0.0214    0.0250 ...
```

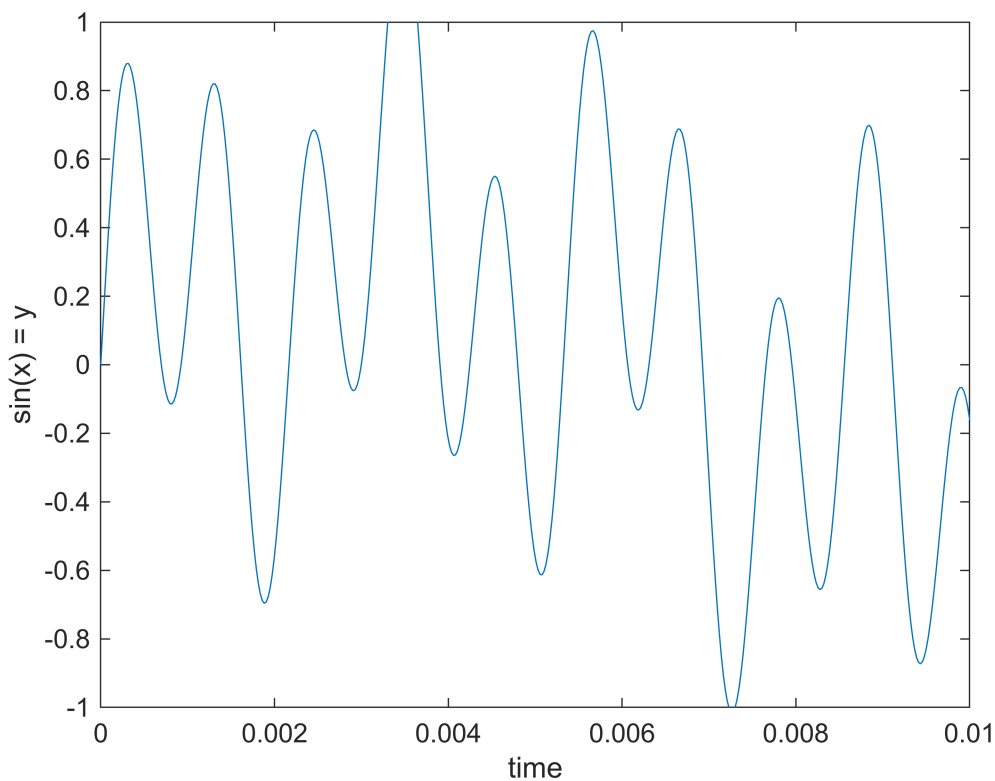
```
plot(x_0,sinY_d2)
xlabel("time")
ylabel("sin(x) = y")
xlim([0 0.03])
ylim([-1 1])
```



%4. Generujte zložený signál ako súčet z predchádzajúcich vygenerovaných signálov.  
 sumSin = sinY\_ais5 + sinY\_fis4 + sinY\_d2 %výsledny signál po ščitání  
 dig. signalov podľa tónov, asi5, tis4, d2

```
sumSin = 1×132301
         0    0.1030    0.2045    0.3031    0.3974    0.4862    0.5682    0.6422 ...
```

```
plot(x_0, sumSin)
xlabel("time")
ylabel("sin(x) = y")
xlim([0 0.01])
ylim([-1 1])
```



```
%5.K zloženému signálu aditívne pripočítajte šum s magnitúdou z rozsahu <0,01:0,05>.
sizeOfSigna = size(sumSin)                                %výpočet počtu časov vzorkovania
```

```
sizeOfSigna = 1x2
             1      132301
```

```
sum = rand(1, sizeOfSigna(2))                             %naplnenie vektora náhodnými
čislami, (s ich počtom rovným počtu vzorkovacích časov), keďže sum berieme ako
náhodnu a nesuvisiacu hodnotu k hodnote zvuku
```

```
sum = 1x132301
      0.7716      0.0415      0.7037      0.9637      0.0654      0.2100      0.8500      0.5048 ...
```

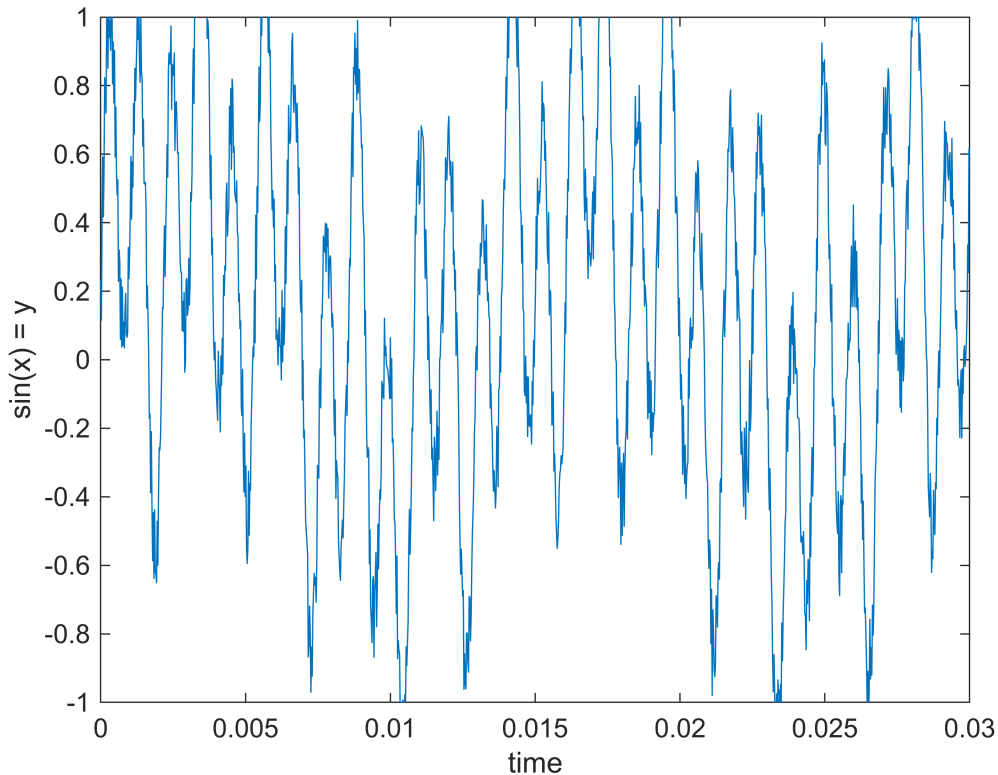
```
sum_mag = mod(rand() + 0.1, 0.5)                          %náhodný výber hodnoty amplitúdy, z
intervalu 0.1-0.5
```

```
sum_mag = 0.3001
```

```
sum = sum * sum_mag                                       %výsledný signal po pridaní šumu do
predošlého súčtu signálov
```

```
sum = 1x132301
      0.2316      0.0125      0.2112      0.2892      0.0196      0.0630      0.2551      0.1515 ...
```

```
plot(x_0, sumSin + sum)
xlabel("time")
ylabel("sin(x) = y")
xlim([0 0.03])
ylim([-1 1])
```



% prehravanie signalov, od jednotlivých tónov, po ich sumu, a nakoniec s pridaním sumu

```
audio1 = audioplayer(sinY_ais5, length(sinY_ais5));  
audio2 = audioplayer(sinY_fis4, length(sinY_fis4));  
audio3 = audioplayer(sinY_d2, length(sinY_d2));  
audio4 = audioplayer(sumSin, length(sumSin));  
audio5 = audioplayer(sumSin + sum, length(sum));  
playblocking(audio1, 3)  
playblocking(audio2, 3)  
playblocking(audio3, 3)  
playblocking(audio4, 3)  
playblocking(audio5, 3)
```

## Úloha 2

Znížte vzorkovaciu frekvenciu u Vami vygenerovaného signálu:

- na polovicu
- na štvrtinu
- na hranicu danú Nyquistovým teorémom
- a aj s porušením Nyquistovho teorému.

Vizualizujte dané signály v jednom grafe. Signál si vypočujte a vyhodnoťte kvalitu.

Pomôcky : funkcia: *downsample()*

```
%opetovné zadanie niektorých základných premenných pre lepší prehľad (a
ovladateľsi spustanie, iba sekcie)
fs = 44100;
freq = 932.24;
x_vec = 0:1/fs:3
```

```
x_vec = 1×132301
    0    0.0000    0.0000    0.0001    0.0001    0.0001    0.0001    0.0002 ...
```

```
%base signal
signal1 = sin(2 * pi * x_vec * freq) * mag_ais5
```

```
signal1 = 1×132301
    0    0.0795    0.1575    0.2328    0.3040    0.3698    0.4291    0.4809 ...
```

```
% na polovicu
x_1 = downsample(x_vec, 2);
newSin1 = downsample(signal1, 2);
% na štvrtinu
x_2 = downsample(x_1, 2);
newSin2 = downsample(signal1, 4);
%výpočet potrebného dividera, pre funkciu downsampling, keďže máme
%základnu hodnotu fs, tak potrebujeme vedieť koľko krát ju mám vidieť
%abych dostal cieľový sampling freq, v tomto prípade pre downsampling
%signálu na hranicu nyquistovho teoremu, tj. cca. 2.1x najvyššia
%frekvencia v signály.
multiOnNyquist = round(fs / (freq * 2.3))
```

```
multiOnNyquist = 21
```

```
%opet výpočet potrebného dividera, ale v tomto prípade pre sample
%frekvenciu ktorá je pod hranicou nyquistovho teoremu.
multiDownNyquist = round(fs / (freq * 1.7))
```

```
multiDownNyquist = 28
```

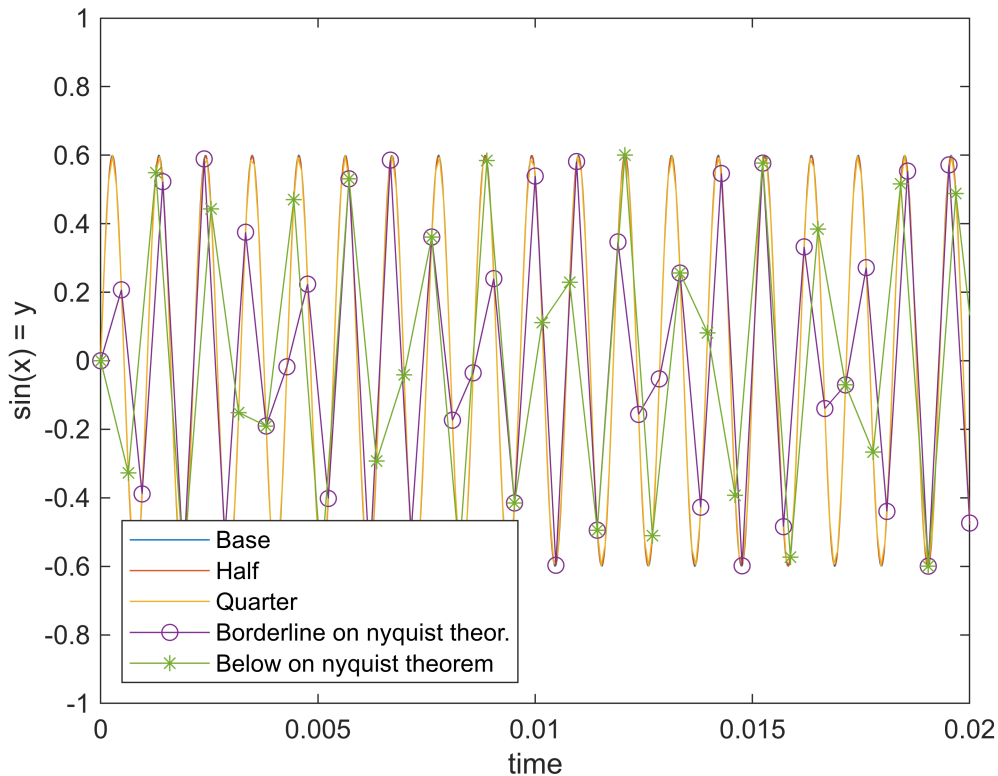
```
% na hranicu danú Nyquistovým teorémom
x_3 = downsample(x_vec, multiOnNyquist);
newSin3 = downsample(signal1, multiOnNyquist);
% s porušením nyquistového teoremu
x_4 = downsample(x_vec, multiDownNyquist);
newSin4 = downsample(signal1, multiDownNyquist);
plot(x_0, signal1)
% Vizualizácia signálov na jednom grafe
hold on
plot(x_1, newSin1)
plot(x_2, newSin2)
plot(x_3, newSin3, "-o")
plot(x_4, newSin4, "-*")
```



```

legend('Base','Half','Quarter','Borderline on nyquist theor.','Below on nyquist
theorem','Location','southwest')
xlabel("time")
ylabel("sin(x) = y")
xlim([0 0.02])
ylim([-1 1])
hold off;

```

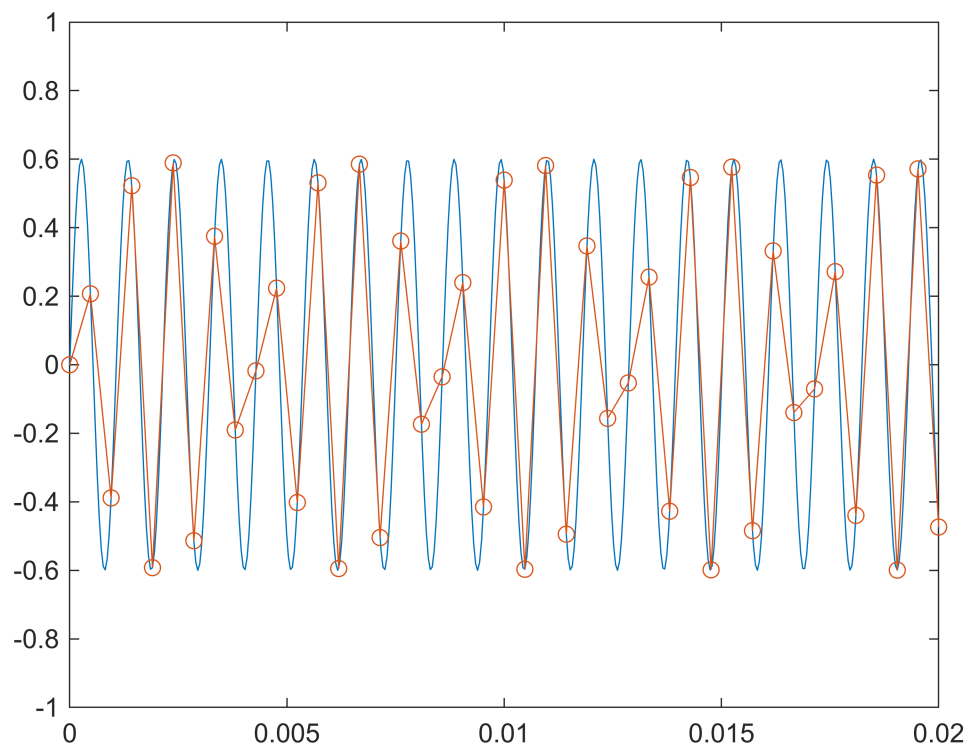


% Vysledok: Grafy ktr. boli podvzorkované na polovicu a stvrtinu,  
 % zachovávajú svoj, priebeh s minimálnou odchylkou od základu, avšak graf  
 % ktorý bol podvzorkovaný na hranicu nyquist. theoremu. tj. 1/21 pôvodného  
 % (vzhľadom na to že vzorkovanie má cca. 2x 930 Hz) vykazuje väčšie  
 % odchylky, ktoré su na hranici podobnosti k pôvodnemu grafu ( výkyvy  
 % amplitúdi/maximá sú v podobnom umiestnená ako base graf), toto je však už  
 % porušené u grafu ktr. nedodržiava nyqst. theroem.

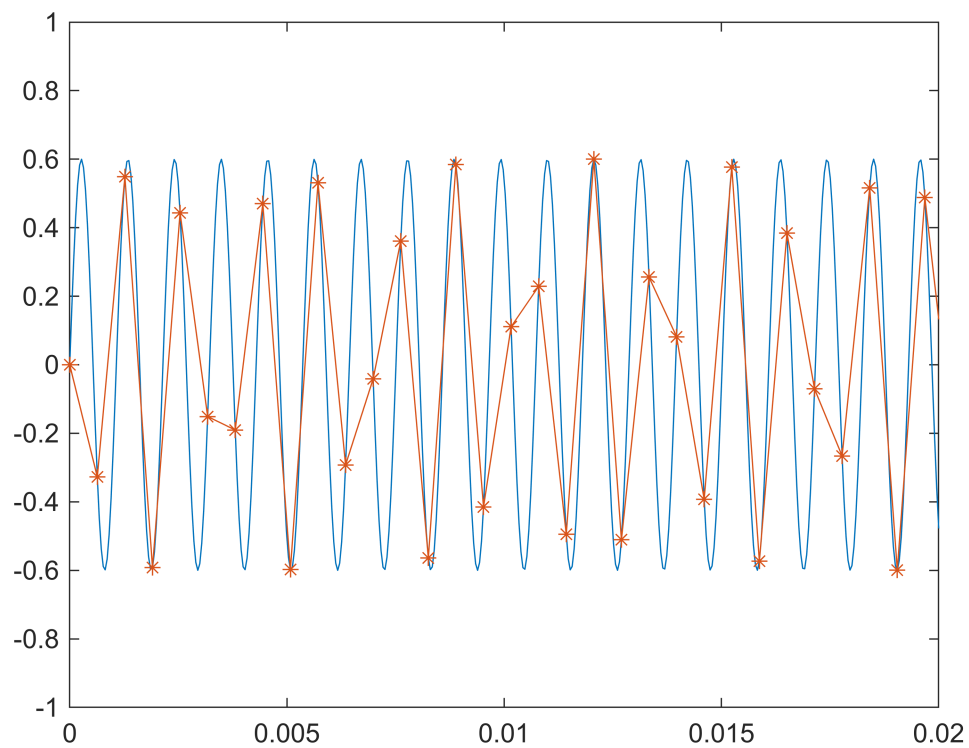
```

plot(x_1, newSin1)
hold on
plot(x_3, newSin3, "-o")
xlim([0 0.02])
ylim([-1 1])
hold off;

```



```
plot(x_1, newSin1)
hold on
plot(x_4, newSin4, "-*")
xlim([0 0.02])
ylim([-1 1])
hold off;
```



```
new_audio0 = audioplayer(signal1, length(signal1));
new_audio1 = audioplayer(newSin1, length(newSin1));
new_audio2 = audioplayer(newSin2, length(newSin2));
new_audio3 = audioplayer(newSin3, length(newSin3));
new_audio4 = audioplayer(newSin4, length(newSin4));
playblocking(new_audio0, 3)
playblocking(new_audio1, 3)
playblocking(new_audio2, 3)
playblocking(new_audio3, 3)
playblocking(new_audio4, 3)
```

```
% Vzhľadom na kvalitu zvuku, tak minimálne prvé tri zvuky, tj. základný
% tón, polovica a štvrtina boli na počutie rovnaké, rozdiel bol
% zaznamenný v 4. signále ktr. je na hranici nyquistovho teoremu a v 5.
% boli zmeny zjávne hlavne vzhľadom na to že celé úseky zvuku boli zdatelne
% nižšieho tónu
%
```