Laboratorijas darba Nr. 4
MAINĪGAS FREKVENCES SIGNĀLA SPEKTRA ANALĪZE
ATSKAITE

Mārtiņš Dundurs
rect0 grupa, apl. nr.

2018. gada 17. janvārī

1 DARBA MĒRĶI

Šī darba mērķis bija izpētīt kā ģenerēt un attēlot laikā mainīgu frekvenču signālu ar vairākiem paņēmieniem - atsevišķās frekvenču un laika telpās 2D grafiku veidā, kopējā frekvenču un laika 3D telpā ar 3D grafiku vai spektrogrammas projekciju. Iemācīties izmantot gatavus optimizētu Furjē transformācijas algoritmu FFT.

2 TEORĒTISKAIS PAMATOJUMS

2.1 Frekvences izmaiņas pēc laika likums

Šajā darbā mēs ģenerēsim signālu, kurš mainās laikā pēc noteikta likuma. Mūsu mērķis ir šo likumu formulēt 4 kārtas polinoma formā:

$$f = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \ ,$$

kur f - frekvence un t - laiks.

2.2 Ātrā furjē transformācija (FFT)

Šajā darbā mēs izmantosim *ātro Furjē transformāciju*. Tā izmantojama tikai diskrētiem signāliem. Tā ir algoritms, kurš izmanto faktu, ka, izdarot DFT, ir jāveic reizināšanas un saskaisīšanas darbības ar vienādiem elementiem vairākas reizes. Šie elementi atrodami pēc noteiktas sakarības. Piemēram, ja mēs izmantotu DFT pēc definīcijas $N = 1024$ nolašu signālam, tad mums būtu jāizdara $O = N^2 = 1048576$ darbības. Bet ar FFT būtu jāveic $O = Nlog_2N = 10240$ operācijas. Visātrākajiem FFT algoritmiem ir nosacījums nolāšu skaitam būt ar bāzi 2, tomēr pastāv lēnāki FFT algoritmi, kuri spēj darboties arī ar citas bāzes nolašu skaitiem.

2.3 Īstermiņa Furjē transformācija (STFT)

STFT - short-time Fourier transform. STFT būtība ir visu signāla laiku sadalīt noteiktā skaitā segmentu un veikt katram segmentam DFT atsevišķi. Tādā veidā mēs varam iegūt virkni spektru katram laika segmentam. Ja mēs šos spektrus aplūkojam attiecīgā secībā, mēs gūsim vienlaicīgi priekšstatu gan par amplitūdas, gan spektra izmaiņām laikā.

3 DARBA GAITA UN REZULTĀTI

3.1 Mainīgas frekvences signāla konstruēšana

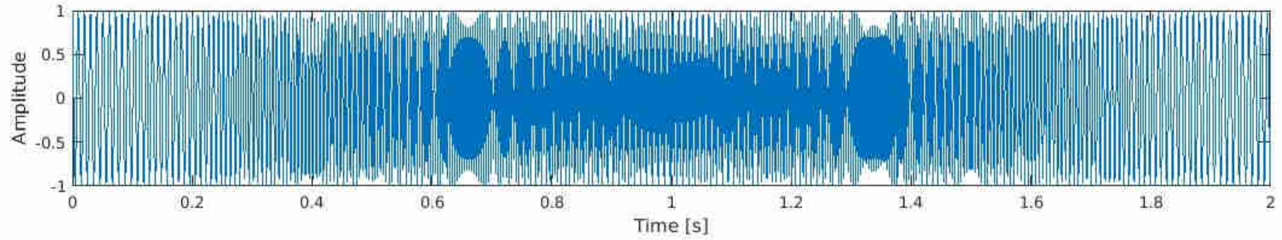
Mums tika doti 5 elementu laika $t0[\,]$ un attiecīgo frekvenču $f0[\,]$ vektori. Lai iegūtu analītisku frekvenšu/laika sakarību mēs šos vektorus interpolējām ar 4 kārtas polinomu un ieguvām a_m koeficientu vektoru `p`.

```
p = polyfit(t0,f0,4);
```

Rezultātā no šī polinoma mēs varam iegūt amplitūdu vērtības ar `chirp()` funkciju. Tā ģenerē kosinusa funkciju ar konstantu amplitūdu mūsu noteiktā 2 $[s]$ intervālā, balstoties uz frekvenci no `p`.

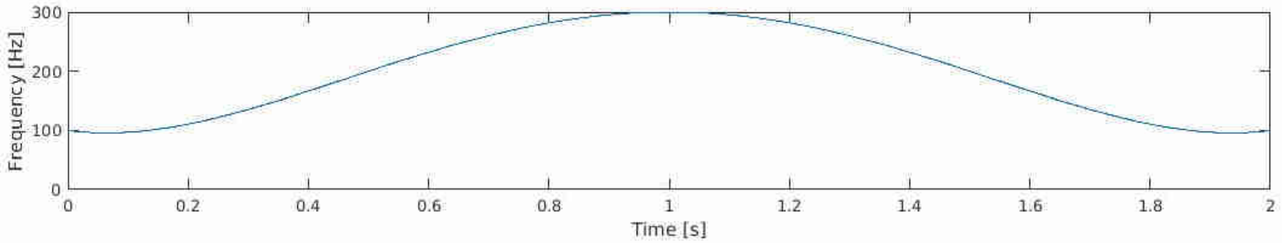
```
y = chirp(t,p);
```

Mēs iegūstam sekojošu signālu:



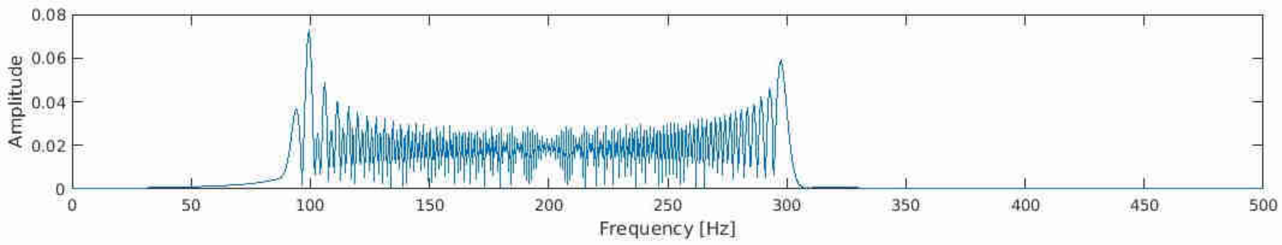
Mēs varam arī atrast noteikto frekvences vērtību katrā laika punktā, izmantojot gatavu funkciju, kas, izmantojot polinoma koeficientus, aprēķina tā vērtību katrā norādītajā punktā. Redzam, ka esam ievērojuši nolašu teorēmu, par cik mūsu frekvences nepārsniedz 500, kas ir puse no izmantotās diskretizācijas frekvences 1 kHz.

```
freq = polyval(p,t);
```



Mums atliek attēlot šī signāla spektru. Priekš tā mums vajadzēt veikt DFT un mēs izmantosim gatavo `fft()` funkciju. Mēs izmantosim tikai nolašu teorēmas nosacījumu apmierinošo frekvenču diapazonu.

```
Y = fft(y)/L;
Y = abs(Y([1:L/2+1]));
F = [0:Fs/L:Fs/2];
```



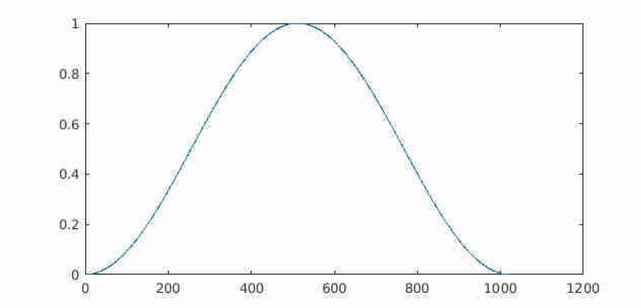
Mēs redzam spektrālo ainu, kas neatbilst harmoniska signālam sagaidāmai. Pie tam arī amplitūdas ir atšķirīgas, kaut arī mēs ģenerējām harmonisku signālu ar konstantu amplitūdu. Iemesls tam ir tas, ka šis spektrs attēlo kopainu visam laika periodam, kurā frekvences bija ar dažādām vērtībām. Spektram ir dažādas amplitūdas, jo tajās vietas, kur mūsu polinoms ir "lēzenāks", frekvences ir stabilākas ilgākā laika posmā. Šī iemesla dēļ šīs frekvences spektrā nes lielāku ieguldījumu un līdz ar to spektrs attēlo lielāku amplitūdu, kaut arī katrā laika momentā visām frekvencēm ir viena un tā pati amplitūda. Papildus tam, DFT veikšanā tiek zaudēta visa informācija par frekvenču secību laikā. Pēc spektra mēs nevaram pateikt vai signāls sākās ar frekvencēm pie aptuveni 300 Hz vai 100 Hz, kaut arī zinot iepriekšējo grafiku mēs zinām ka sākotnējās frekvences bija ap 100, intervāla vidū tās sasniedza aptuveni 300 un beigās atgriezās pie 100 Hz. Turmpākajā daļā mēs aplūkojām algoritmu, kas ļāva mums attēlot spektru, saglabājot aptuvenu informāciju par frekvenču izmaiņām laikā.

3.2 Slīdošā loga DFT transformācijas aprēķins un spektrogrammas konstruēšana

Šeit mēs konstruējām STFT algoritmu. Ka segmenta garumu mēs izvēlējāmies 128 nolases ar 8 nolašu soli. Tas nozīmē, ka blakus segmenti savā starpā pārklājas ar 120 nolasēm. Pēc palīgmainīgo izveidošanas

```
M = floor((L-LapNo)/SmplNum);
Mz = L-SmplNum;
yz = [y, zeros(1,Mz)];
ySig = zeros(NFFT,M);
```

mēs definēsim logu, kurš modificēs katrā segmenta vērtības tā, lai lielākā nozīme būtu segmenta vidējām vērtībā. Mēs izmantosim `hanning()` funkciju, kuras forma ir sekojoša.



Pēc tam mēs cikliski pareizinājām ar Hanna funkciju katru segmentu.

```
for k = 1:M-1
    idxCol = (1:NFFT)+(k-1)*SmplNum;
    ySig(:,k) = yz(idxCol).*wnd;
end
```

Rezultātā ieguvām svērtus segmentus, kuriem tālāk veicām FFT katram atsevišķi.

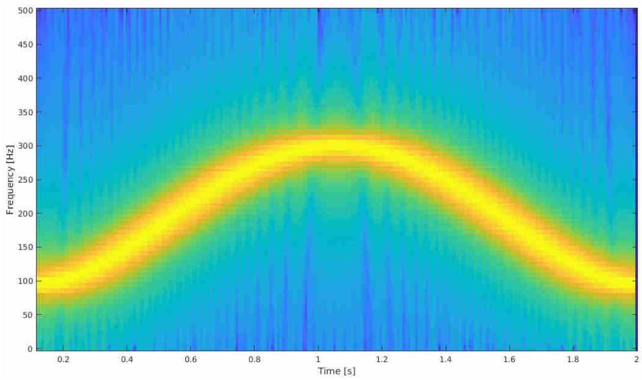
```
S = fft(ySig);
S = mag2db(abs(S(1:NFFT/2+1,:)));
```

Tagad lai attēlotu šos spektrus spektrogrammā, mēs izmantosim nu jau citu laika intervālu ar soli 8. Līdzīgi būs vajadzīgs arī jauns Frekvenču diapazons.

```
T = t(LapNo:SmplNum:L);
F = [0:Fs/NFFT:Fs/2];
```

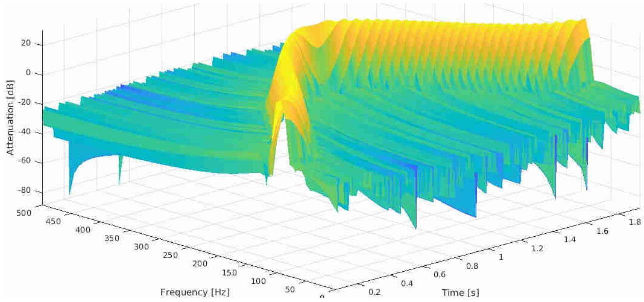
Tagad varam attēlot spektrogrammu ar funkciju `imagesc()`.

```
imagesc(T,F,S);
axis xy;
```

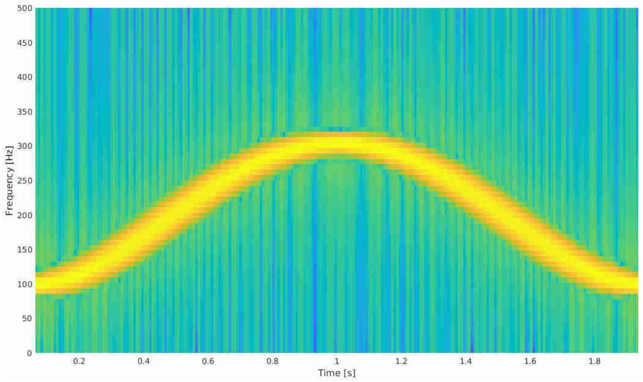


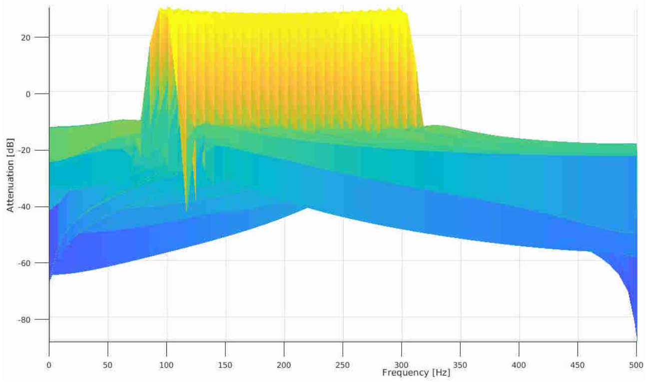
Šajā attēlā krāsu temperatūra norāda uz frekvenču amplitūdu. Tātad spektrs šeit ir attēlots krāsu dimensijā. Nākamais mūsu solis bija izveidot 3D spektrogrammu. Šeit mēs spektra amplitūdu attēlojām deciBelos.

```
[S,F,T]=spectrogram(y,NFFT,LapNo,NFFT,Fs);
S = mag2db(abs(S));
surf(T,F,S,'edgecolor','none'); axis xy; axis tight
```



Šo 3D attēlu varam pagriezt, lai attēlotu $f(t)$ un $Y(f)$ grafikus. Pēc šī attēla mēs varam saprast ka katrā laika segmentā frekvenču spektrs ir harmoniska signāla spektrs ar noteiktu frekvenci. Proti, katrā laika momentā spektrā ir tikai viena kosinusoīdas komponente.





Acīmredzama ir grafiku analoģija ar darba sākumā iegūtajiem. Tiesa, jāpiezīmē ka $Y(f)$ grafikā vairs nav redzamas dažādās frekvenču amplitūdas, kuras nepatiesi tika attēlotas 2D grafikā.

4 SECINĀJUMI

Šī darba galvenā mācība mums bija tā, ka veicot FT mēs zaudējam informāciju par frekvenču izmaiņām laikā. Lai mēs to nezaudētu, mēs varam izmantot STFT algoritmu un iegūt spektrogrammu, kurā redzamas ne tikai frekvenču amplitūdas/vājinājums atkarībā no frekvencēm, bet arī frekvenču izmaiņas atkarībā no laika.

Tāpat mēs šajā darbā apguvām kā izmantot gatavus DFT algoritmus, kas ir ievērojami ātrāki kā DFT rēķināšana pēc definīcijas.

5 PIELIKUMS

Pielikumā fails:

`lab04.m`