* Short and to the point (max 10 pages of text - excluding graphs, tables, formulas etc)
* show software design and (filter) calculations
* show workflow structure and planning
* discuss used algorithms and show the theory+ formulas!
* discuss implementation in software (using Python)
* discuss the learning Goals from ch. 0.1 of the study guide
* discuss the research questions from the above assignment
* add a schematic overview of the contribution of each student individually

1. **Try to filter the audio-recordings such that only clear noise free bird songs remain. Use for this purpose the discrete filtering techniques as taught in the DSP-course.**

Om de achtergrond geluiden weg te filteren moet het frequentiespectrum van elke sample bekeken worden. Zodra het bekend is in welke frequenties het achtergrondgeluid zich bevindt is het mogelijk om deze weg te filteren. Dit wordt gedaan door eerst een fast-fourier transformatie uit te voeren waardoor er per sample voor elke frequentie een amplitude wordt berekend. Hierdoor is het mogelijk om deze amplitude ook gelijk te stellen aan 0, en daardoor de audiodata in deze frequentie weg te filteren. Omdat dit voor elke sample in het hele file gedaan moet worden is hier code voor geschreven die dat geautomatiseerd doet.

leftFourier = np.fft.rfft(l\_data)  
  
# Kantelfrequenties voor filters. Gelden voor links en ook voor rechts indien aanwezig.  
# single bird  
fk\_hp1 = 5000  
fk\_lp1 = 7500  
#multiple bird  
#fk\_hp1 = 2200  
#fk\_lp1 = 15000  
  
# Filters linker kanaal  
leftFourier = HighPass(leftFourier, fk\_hp1)  
leftFourier = LowPass(leftFourier, fk\_lp1)  
#leftFourier = BandSper(leftFourier, fk\_bp1l, fk\_bp1h)  
  
#data voor FFT plot linker kanaal  
if num == max\_value\_frame:  
 print("SAMPLE LEFT!")  
 left\_frequency\_prefilter\_data = np.fft.rfft(l\_data)  
 left\_frequency\_postfilter\_data = leftFourier  
  
# Inverse FFT left channel  
new\_l\_data = np.fft.irfft(leftFourier)

Deze code is zo opgebouwd dat de filters eigen functies zijn en doormiddel van het meegeven van een kantelfrequentie is het mogelijk dat deze voor meerdere frequenties hergebruikt kunnen worden.

De filter functies zijn hieronder afgebeeld.

Het filteren wat hier gebeurt is het gelijkstellen aan 0 van de frequenties die weg gefilterd moeten worden. Hierdoor wordt de amplitude van deze frequentie op dat moment 0. Door dit bij elke sample te doen worden al deze frequenties weg gefilterd.

def HighPass(Fourier, fk):  
 # Stel alle frequenties tussen 0Hz en de kantelfrequentie gelijk aan 0  
 Fourier[:fk] = 0  
 return Fourier  
  
  
def LowPass(Fourier, fk):  
 # Stel alle frequenties tussen 20kHz en de kantelfrequentie gelijk aan 0  
 Fourier[fk:] = 0  
 return Fourier  
  
  
def BandSper(Fourier, fkLow, fkHigh):  
 # Stel alle frequenties tussen de kantelfrequenties gelijk aan 0  
 Fourier[fkLow:fkHigh] = 0  
 return Fourier

Om de verdere werking van de software duidelijker toe te lichten is er een flowchart gemaakt welke dit laat zien. Om de leesbaarheid te bevorderen is deze op de volgende pagina geplaatst. Ook staat het in de bijlage als grote afbeelding.

Diagram, schematic

Description automatically generated

Figuur 1 Main programma

Diagram, schematic

Description automatically generated

Figuur 2 Losse funties

1. **Discuss the pro’s and con’s of the filtering techniques applied for the bird song(s)**
2. **One of the recordings contains mulitple birds. Think of ways if it is possible to extract a single bird song**

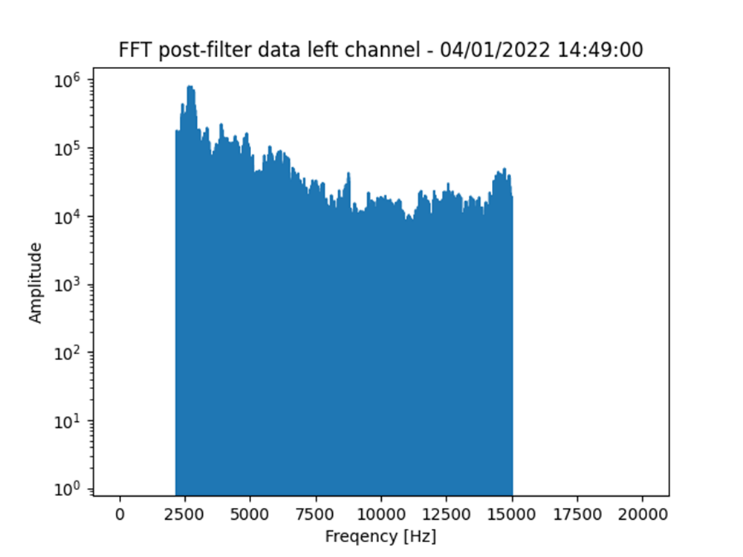
Dit kan gedaan worden door FFT toe te passen op de audiofile met meerdere vogels. Hierin moet je zoeken naar een frequentie van een vogel. Deze frequentie gooi je door een band-pass filter over de volledige audiofile met meerdere vogels. Hierbij zou je alleen de vogel moeten horen die door het band pass filter gefilterd is.

1. **Detect the kind(s) of singing birds you hear (*there are lots of websites containing examples of bird songs*)**

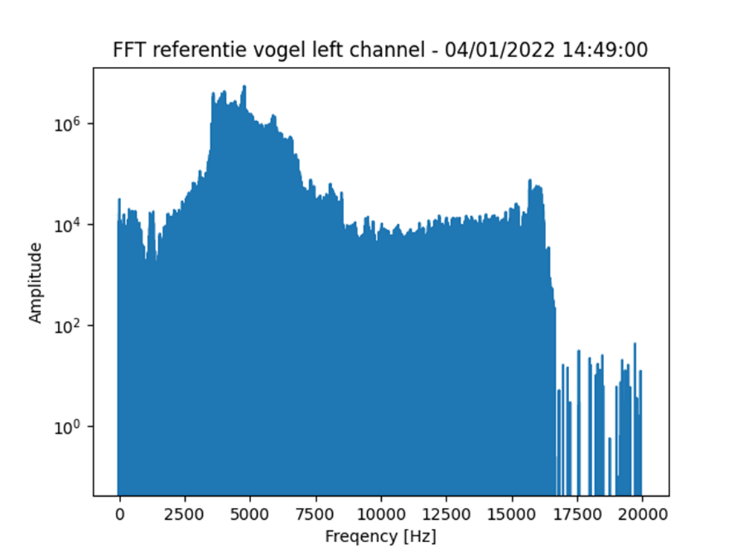
Om te kunnen detecteren wat voor soort vogels te horen zijn in de gegeven audiobestande, hebben we gebruik gemaakt van de BirdNerd app. Deze app kan aan de hand van een audiobestand detecteren wat voor soort vogels er te horen zijn. Wanneer we de gefilterde bestanden in de app gebruiken werden de volgende vogels herkend:

* Tureluur
* Zanglijster

1. **After filtering, apply Fourier spectral analysis to the individual bird songs of the recordings and compare this with recordings you find in the internet on which you also apply spectral analysis.**

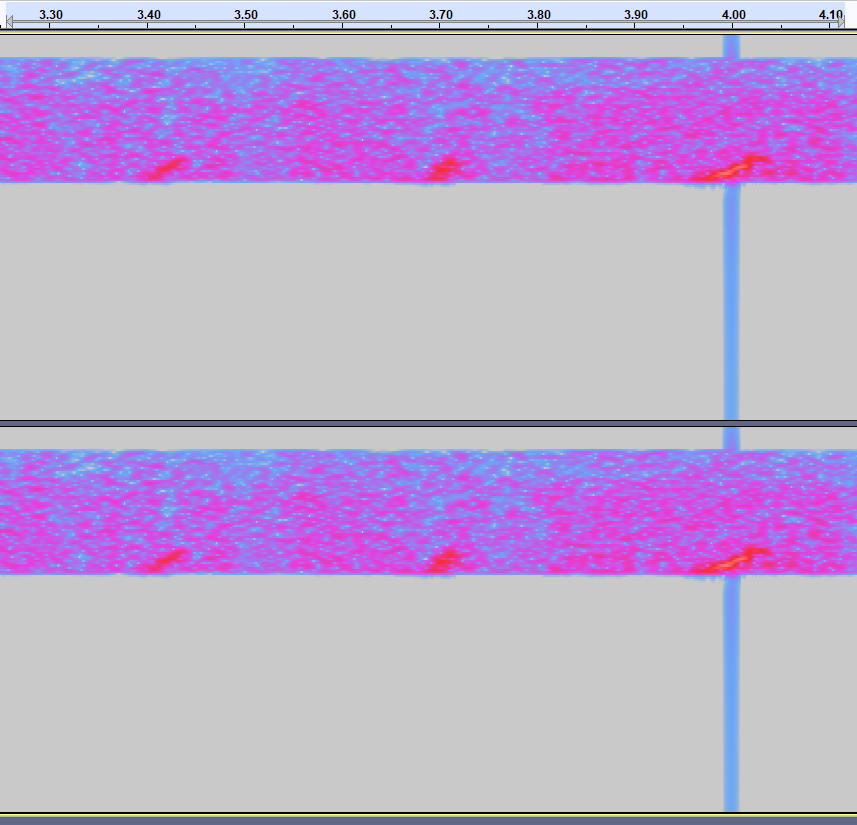
In **Figuur 1** hieronder is een frequentiespectrum analyse van een individuele vogel te zien, na dat het audiobestand gefilterd is.

Figuur 3, Frequentiespectrum analyse van gefilterd audiobestand.

Vervolgens is in **Figuur 2** een frequentiespectrum analyse van een Tjif Tjaf afkomstig van het internet te zien.

Figuur 4, Frequentiespectrum analyse van een Tjif Tjaf

1. **Discuss if ‘spectral recognition’ can be used to obtain the kind(s) of bird(s) from a recording. Is it unique? Focus on why it can or can not from a technical point of view.**
2. Het gebruiken van frequentie spectrum analyse is een goede methode om de verschillende vogels te identificeren. Het idee hierbij is dat voor elke frequentie de intensiteit (amplitude) wordt weergeven door een kleur die “harder” wordt. Dit resulteert in een zogenaamde waterval frequentie analyse. Deze plot combineert de data die verkregen kan worden met een FFT uit alle samples en zet deze in de tijd weer achter elkaar. Dit resulteert in een kleurig spectrum met daar in 3 verschillende data, namelijk het frequentiespectrum, de tijd en de amplitude per frequentie. De afbeelding hieronder is de waterval plot van de “single birds” audio file.



1. De drie rodere vlekken zijn drie tsjilpen van de Tureluur die in deze audiofile te horen is. De rodere, intensere kleur geeft aan dat de amplitudes in deze frequentie hoger zijn dan de omliggende roze en blauwe frequenties. Zoals te zien is, is de frequentieband tussen 0 en 5kHz volledig leeg. Dit is het werk van het highpass filter. Boven de 7.5kHz is ook alles gefilterd en dit is het werk van het lowpass filter. De horizontale as is de tijd, en daardoor is ook de lengte en de interval van de tsjilpen van de tureluur te zien. Dit maakt dat er gebruik gemaakt kan worden van deze manier van data weergeven om de geluiden van vogels met elkaar te vergelijken.

# Taakoverzicht

Om ervoor te zorgen dat de opdracht soepel verloopt zijn de taken onderling verdeeld. Dit komt terug in het taak overzicht hier onder. Bij sommige taken heeft iedereen een aandeel gehad zoals bijvoorbeeld het verslag.

|  |  |  |
| --- | --- | --- |
| Taakoverzicht DSB Opdracht | | |
| ***Datum:*** | ***Taak:*** | ***Gemaakt door:*** |
| 13-dec-21 | Opbouw Python Programma met betrekking tot files importeren/exporteren enz. | Bart |
| 15-dec-21 | Filters uitzoeken en toepassen audio | Dirk-Jan, Bart |
| 15-dec-21 | Plotten audiobestanden voor en na | Anne |
| 22-dec-21 | Fourier Spectral analysis toepassen | Dirk-Jan |
| 3-jan-21 | Opbouw verslag | Dirk-Jan, Anne, Bart |
| 4-jan-21 | Presentatie maken | Dirk-Jan, Anne, Bart |

# Workflow

Om ervoor te zorgen dat de bestanden uitgelezen kunnen worden is er eerst gekeken hoe de wav files met de vogelgeluiden ingeladen kunnen worden. Daarna is het natuurlijk ook van belang om het aangepaste bestand weer terug te exporteren om zo vervolgens verandering op te kunnen merken. Dit hebben we eerst gedaan zonder filters om zo er zeker van te zijn dat het exporteren werkt. Na dat dit werkend is kan er gekeken worden hoe de ruis over de bestanden weg gehaald kan worden zodat de vogelgeluiden duidelijk verstaanbaar zijn. Hierbij zal er gekeken worden naar een plot van de gefilterde audiofiles. Waarna de geëxporteerde audio geen ruis meer bevat door verschillende filters toe te passen wordt de geëxporteerde audio vergeleken met verschillende vogelgeluiden om zo tot een vogelsoort te komen.

* Demonstrate and explain correct operation and implentation at the scheduled time
* Hand in report + software before the deadline
* Do this via uploading Blackboard DSP assignment (or via email)
* Indicate project group number + names in the title