

Fjerning av pipetone

Tittel: Fjerning av pipetone

Forfatter: Andreas Lindeman

Versjon: 1.9

Dato: 05.03.2024

Innhold

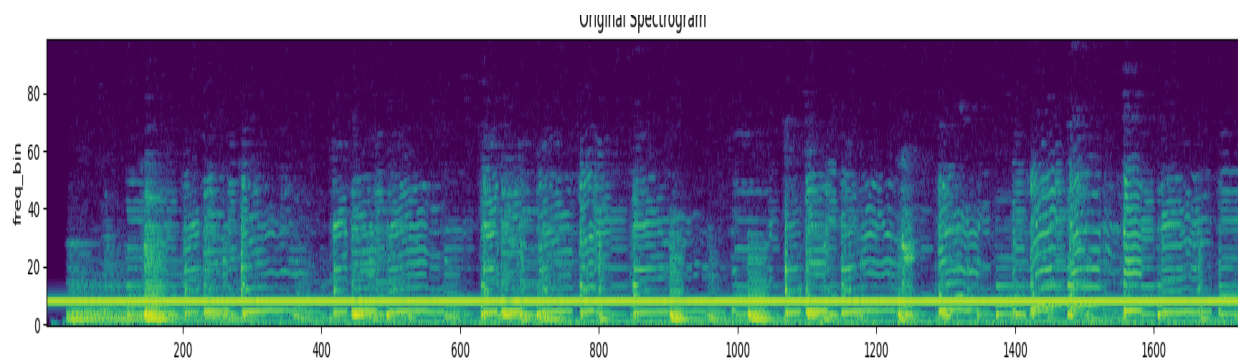
1. Problemstilling	1
2. Prinsipiell løsning	2
3. Realisering og test	4
4. Konklusjon	7
Referanser	7

Problemstilling

I denne oppgaven ble vi presentert for et problem der en uønsket pipetone har infiltrert en lydfil. Denne lydfilen, som inneholder et musikksignal, har blitt korrumpert av en konstant frekvens pipetone. Dette ødelegger den opprinnelige lyd kvaliteten, noe som skaper et behov for en løsning som kan gjenopprette signalets renhet.

Ved å visualisere lydfilen i et spektrogram, med tid på x-aksen og frekvens på y-aksen, kan vi tydelig identifisere tilstedeværelsen av pipetonen. Dette gir oss en visuell representasjon av problemet, og understreker behovet for en effektiv løsning for å fjerne denne uønskede tonen.

Målet vårt er derfor å utvikle et system som kan isolere og fjerne denne pipetonen fra lydfilen, og dermed forbedre signalkvaliteten. Dette vil kreve en grundig forståelse av lydprosesseringsteknikker, samt evnen til å implementere disse teknikkene på en effektiv måte. Figuren under viser lydsignalet som et spektrogram.



Figur 1: Lydsignal som spektrogram, frekvens på y-akse og tid langs x-aksen, den grønne horisontale linjen nesten nederst er grunnet uønsket pipetone i lydsignalet

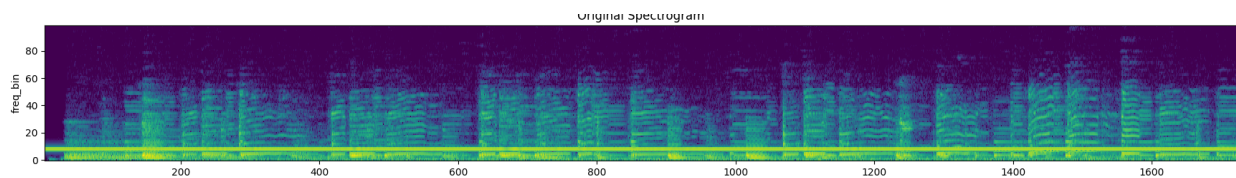
Prinsipiell løsning

Gitt den klare identifikasjonen av pipetonen i spektrogrammet, kan vi utnytte denne informasjonen for å fjerne den fra lydfilen. Selv om denne tonen er en konstant del av dataene, er den relativt liten i forhold til det totale spekteret av frekvenser i musikksignalet. Derfor vil fjerning av denne tonen ikke føre til at den resulterende lyden høres merkelig ut. Vi har identifisert to hovedmetoder for å oppnå dette:

Fjerning av frekvenser

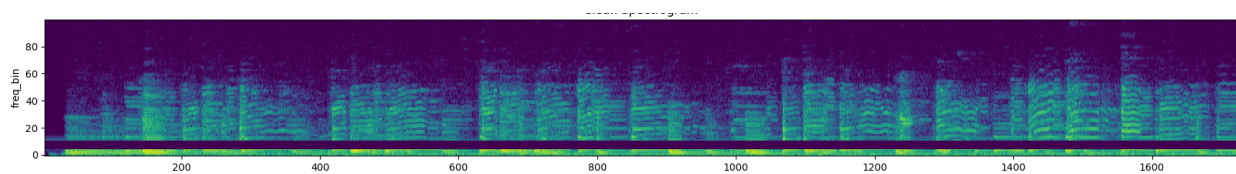
Den første metoden innebærer å fjerne de spesifikke frekvensene som tilsvarer pipetonen direkte fra spektrogrammet. Dette kan betraktes som en form for frekvensfiltrering, liknende et båndstopfilter, hvor vi tillater alle frekvenser unntatt en viss frekvens (dvs., alle frekvenser unntatt pipetonen) å passere gjennom. Den gjenværende daten vil da tilsvare all musikken som ikke ligger i samme frekvens som det pipetonen var.

Med andre ord, vi fjerner all dataen i frekvensene i og rundt der pipetonen befinner seg. I vårt tilfelle vil dataen som vi starter med se slik ut:



Figur 2.1: Originale lydsignalet som spektrogram, frekvens på y-akse og tid langs x-aksen

Vil se slik ut:



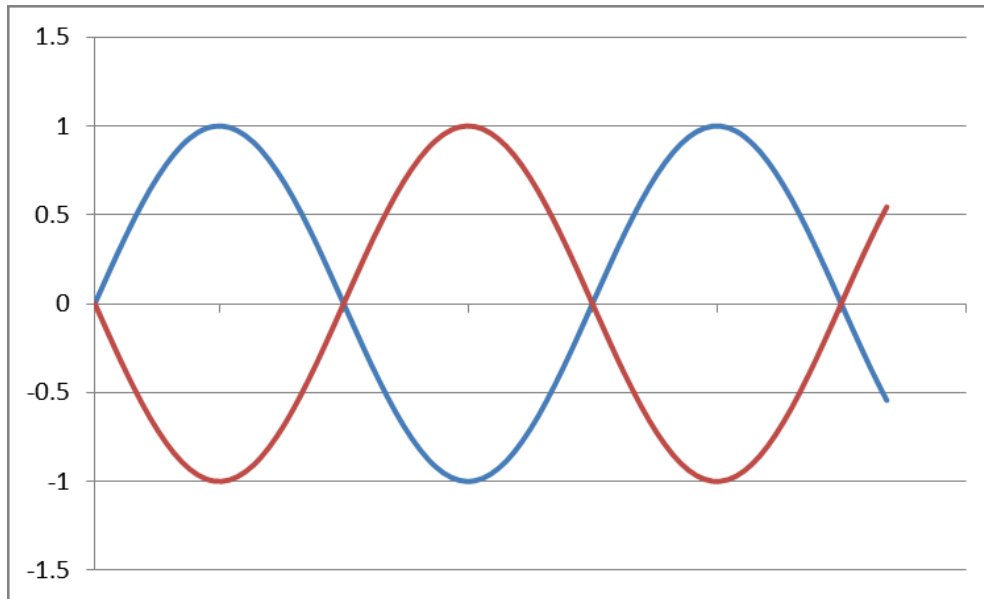
Figur 2.2: Eksempel på behandlet lydsignal som spektrogram

Vi kan i figur 2.2 se at den grønne horisontale linjen som representerer pipetone i lydsignalet nå er fjernet. Den gjenværende daten vil nå bare være ønsket data.

Metode 2 - Legge til en motsatt sinuskurve

Den andre metoden utnytter det faktum at pipetonen bare er en sinuskurve som er lagt til musikken. Ved å legge til en annen sinuskurve som er forskjøvet med 180 grader i forhold til pipetonen, kan vi effektivt kansellere ut pipetonen. Dette skyldes at summen av to sinusbølger med samme frekvens, men forskjøvet med 180 grader, er null. Dette kan resultere i en teoretisk perfekt fjerning av pipetonen. Under ser du en visualisering av to sinusbølger, med 180 grader forskyvning mellom seg:

Fjerning av pipetone



[1] Figur 3: To sinusbølger forskjøvet 180 grader fra hverandre

Begge disse metodene har sine fordeler og ulemper, og valget mellom dem vil avhenge av de spesifikke kravene og begrensningene i vår brukssituasjon. Uansett hvilken metode vi velger, er målet vårt å gjenopprette kvaliteten på det opprinnelige musikksignalet ved å fjerne den uønskede pipetonen.

Realisering og test

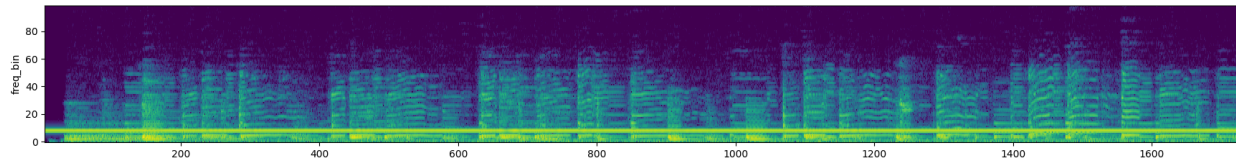
For å løse problemet med pipetonen, er begge de foreslåtte metodene implementert: fjerning av frekvenser og å legge til en motsatt sinuskurve.

For den første metoden, fjerning av frekvenser, ble en algoritme utviklet for å identifisere og fjerne den spesifikke frekvensen av pipetonen fra spektrogrammet. Dette ble oppnådd ved å bruke en form for frekvensfiltrering, liknende et båndstoppfilter. Etter at skriptet har identifisert frekvensen med mest data (midtpunktet i pipetonen), fjerner den gjennomsnittsverdien fra den frekvensen og hosliggende frekvenser påvirket av pipetonen.

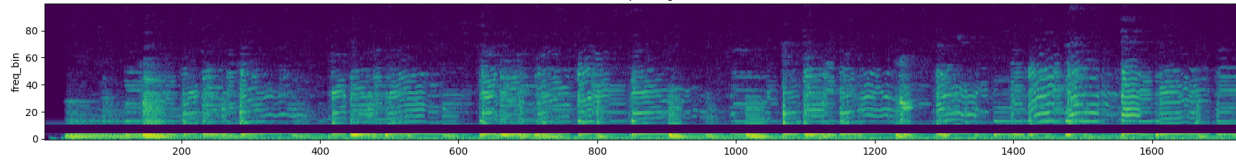
Dette gjør vi ettersom vi da er sikre på at hele pipetonen blir fjernet selv med usikkerheten som kommer ved å endre en lydfil til et spektrogram og tilbake. Under ser du den originale lyden visualisert i et spektrogram, etterfulgt av et spektrogram hvor dataen i frekvensene er helt

Fjerning av pipetone

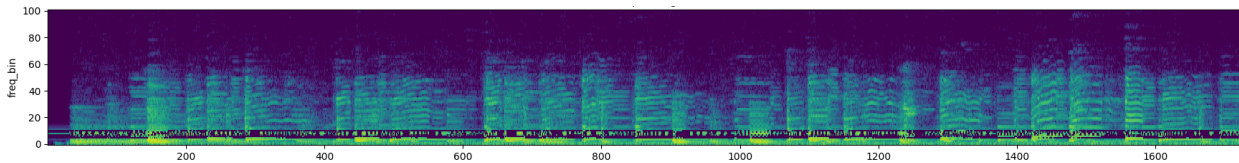
fjernet, og til slutt et spektrogram hvor dataen er best mulig prøvd bevart.



Figur 4.1: Originale lydfile som spektrogram

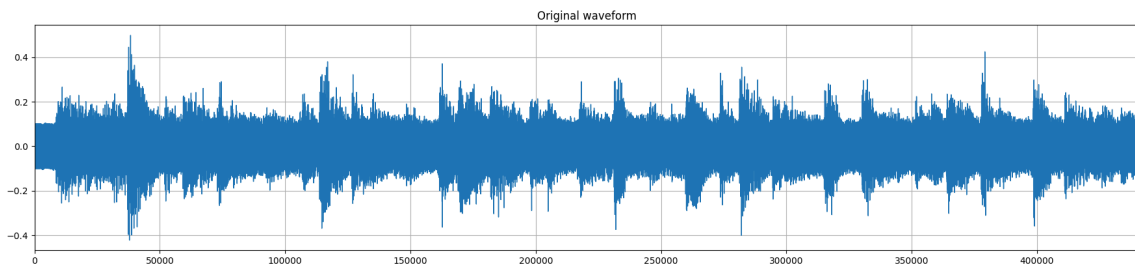


Figur 4.2: Lydfile med all data påvirket av pipetonen fjernet

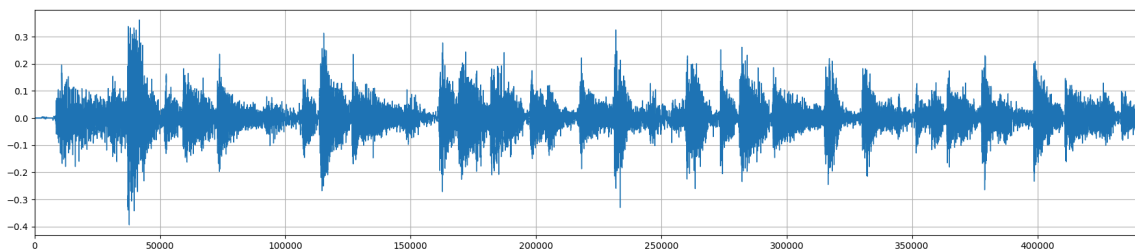


Figur 4.3: Lyddataen med dataen i pipetonen best mulig prøvd bevart

Her kan vi tydelig se på det nederste spektrogrammet at mye data har blitt fjernet. Dette er veldig tydelig på spektrogrammet, men lyden generert av scriptet høres fortsatt naturlig ut. Under ser du den originale lyden samt lyden generert av spektrogrammet.



Figur 5.1: Originale lydfile visualisert på bølgeform



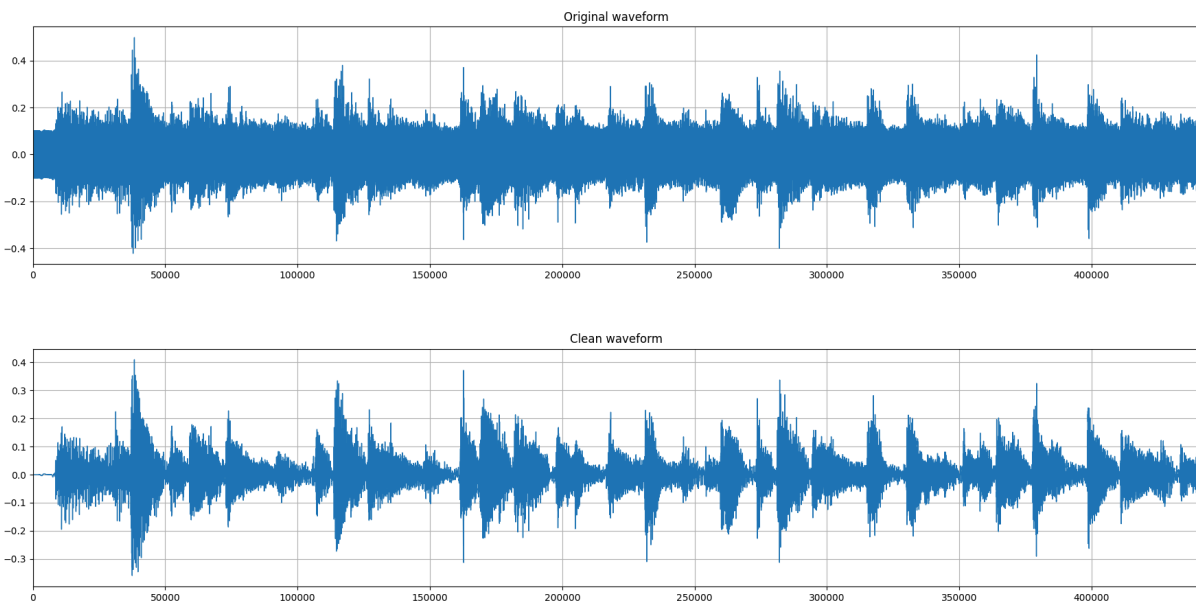
Figur 5.2: Den prosesserte lyden visualisert på bølgeform

Fjerning av pipetone

Det er tydelig at den genererte lyden har betydelig mindre støy enn den originale lyden.

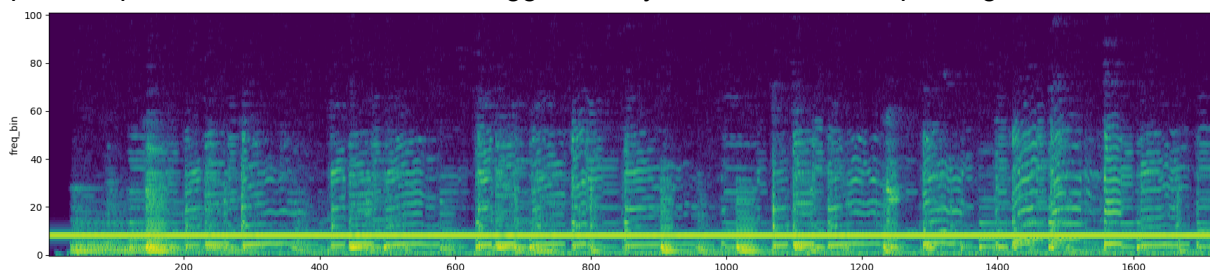
For den andre metoden; legge til en motsatt sinusurve, ble en annen algoritme utviklet. Denne algoritmen finner først ut amplitude, frekvens og forsyvning til sinusbølgen som representerer pipetonen. Deretter genererer den en sinusurve, med samme amplitude og frekvens, men som er forskjøvet med 180 grader i forhold til pipetonen, og legger den til lydfilen. Dette resulterer i at pipetonen effektivt blir kansellert ut.

Under ser du en visualisering av lyden før og etter den har blitt behandlet av denne algoritmen:



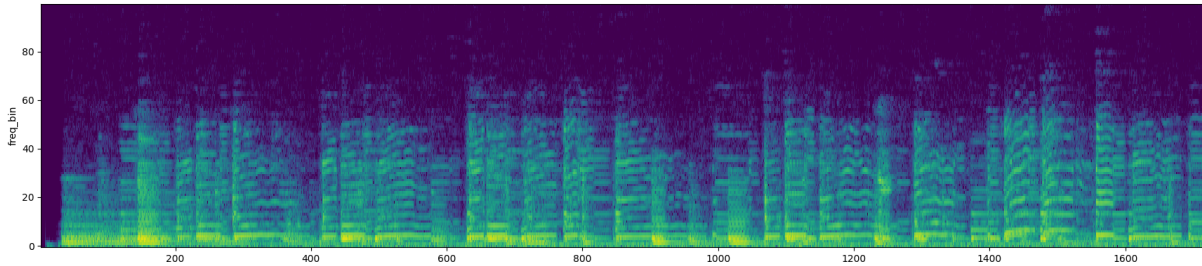
Figur 6: Lyddata på bølgeform etter forskjøvet sinusurve er lagt til

Det første å legge merke til her er den minimal endring i lydsignalet utenfor støysonen. Dette er spesielt åpenbart hvis vi visualiserer begge de to lydfilene i hvert sitt spektrogram.



Figur 7.1: Spektrogram av den originale lydfilen

Fjerning av pipetone



Figur 7.2: Lyddataen med en 180 grader forskjøvet sinusurve

På det nederste spektrogrammet i figuren over er det ikke funnet spor av pipetonen, samt spor etter at dataen har vært prosessert av et program. Dette indikerer at lyden mest sannsynlig er ekstremt nær den originale lyden. Dette resultatet er nettopp det vi ønsket å oppnå med vårt system.

Etter implementeringen, ble begge metodene grundig testet for å sikre deres effektivitet. Testene inkluderte både kvalitative vurderinger, som å lytte til den resulterende lyden, og kvantitative målinger, som å analysere det resulterende spektrogrammet.

Begge metodene viste seg å være effektive i å fjerne pipetonen og forbedre kvaliteten på musikksignalet.

Konklusjon

Etter grundig analyse og testing av begge metodene for å fjerne pipetonen fra lydfilen, har vi kommet til en konklusjon. Hvis pipetonen er konstant, er den mest effektive metoden å legge til en motsatt sinusurve. Dette gir en renere lydfile uten den uønskede pipetonen.

Selv om fjerning av frekvenser også er en effektiv metode, har vi funnet at legge til en motsatt sinusurve gir en mer naturlig og mindre forvrengt lyd. Dette skyldes at denne metoden ikke fjerner noen del av det opprinnelige spekteret av frekvenser i musikk signalet, men heller nøytraliserer den uønskede tonen.

Derfor, gitt en uønsket konstant pipetone, vil å legge til en motsatt sinusurve være den beste metoden for å forbedre lyd kvaliteten.

Referanser:

[1] Figur 3 hentet fra *producelikeapro*:

https://producelikeapro.com/blog/wp-content/uploads/2018/07/What-Is-Phase-Cancellation_2.png

Fjerning av pipetone

Kode for å visualisere data inspirert av *Pytorch example code*:

<https://pytorch.org/audio/main/generated/torchaudio.transforms.Spectrogram.html>