

1 Opgave 5 - Nedsample signal 1 med faktor 4

I følgende opgave skal jeg tage S_1 og nedsample det med en faktor 4. Hertil bruger jeg funktionen *decimate()*, hvor argumenterne er listen af samples fra S_1 og derefter den faktor som det skal nedsamples med.

Listing 1: Nedsampel med faktor 4

```
1 y(5).sample = decimate(y(1).sample , 4);
```

Hvis man herefter plotter det får man som på Figure 1

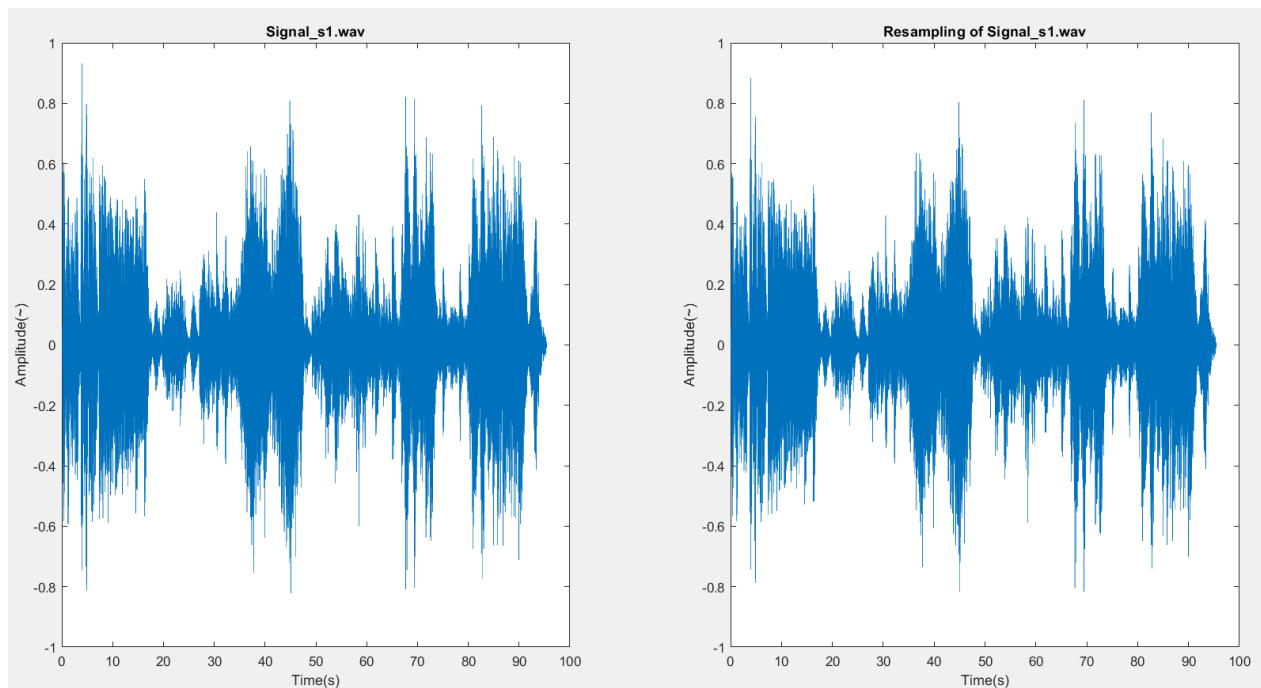


Figure 1: Downsampling af signal 1

2 Opgave 6 - Beskriv forskellen på signalet fra opgave 5

Ud fra plottet i Figure 1 kan man se der er nogle små ændringer, specifikt lige omkring 70 sekunder, hvor man kan se, at det resamplede signal ikke får samme højde som det originale. Derudover kan man også høre, at selvom signalet holder sin integritet, mister den en stor del af dybden og kvaliteten af lyden.

3 Opgave 7 - Sammenlign de to kanaler i signal 2

I denne opgave skal man sammenligne de to kanaler i S_2 og beskrive forskelle og ligheder.

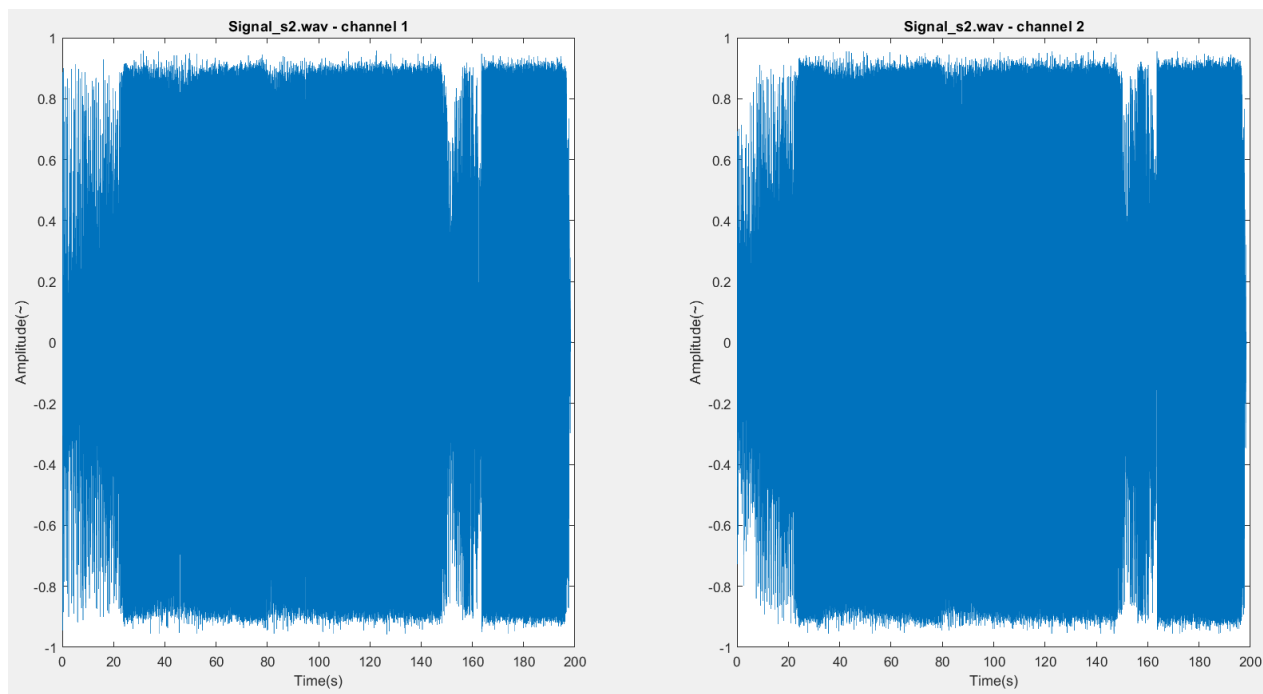


Figure 2: Begge kanaler af Signal 2

På Figure 2 kan man se, at kanal 1 starter højt, hvorimod kanal 2 starter lavt og så kommer op på samme amplitude som kanal 1 når man kommer lidt længere ind i signalet.

4 Opgave 8 - Kvantisering af signal 2(venstre) med 4 bit

I denne opgave skal man tage venstre side af signal 2 og kvantisere det til 4 bit. Hertil har jeg brugt den funktion vi har fået givet *quantizeN()*. Denne funktion tager en liste af samples og så den mængde bits som man vil kvantisere den til. Efter at kvantisere til 4 bit kan man plotte det og få resultatet som ses på Figure 3

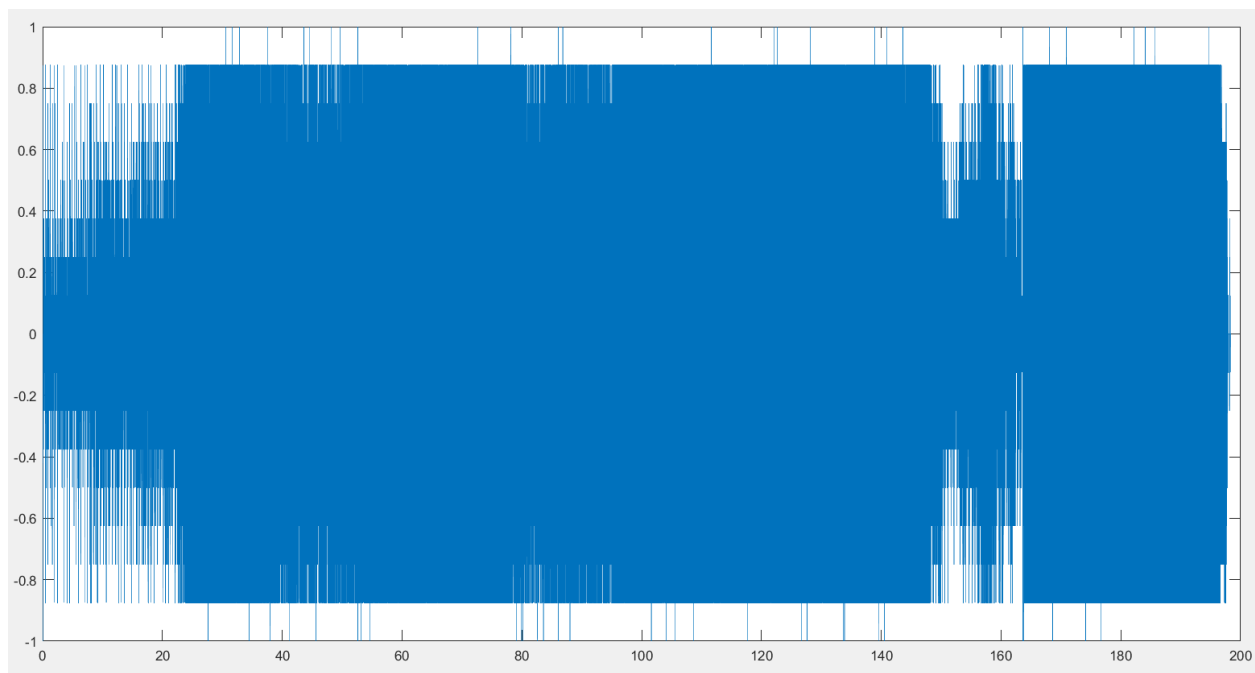


Figure 3: 4 bit kvantisering af signal 2's venstre kanal

Til sammenligning med Signal 2's venstre kanal på Figure ?? kan man se, at der er meget mere detalje på det originale signal. Dette giver god mening efter vi har gjort det kvantiserede signal's bredde til 4 bit. Derudover når man afspiller det kvantiserede signal lyder det bare som radio støj.

5 Opgave 9 - Lav fade out på signal 3

I denne opgave skal man lave et eksponentielt fald til 5% over den sidste tredjedel af signal 3.

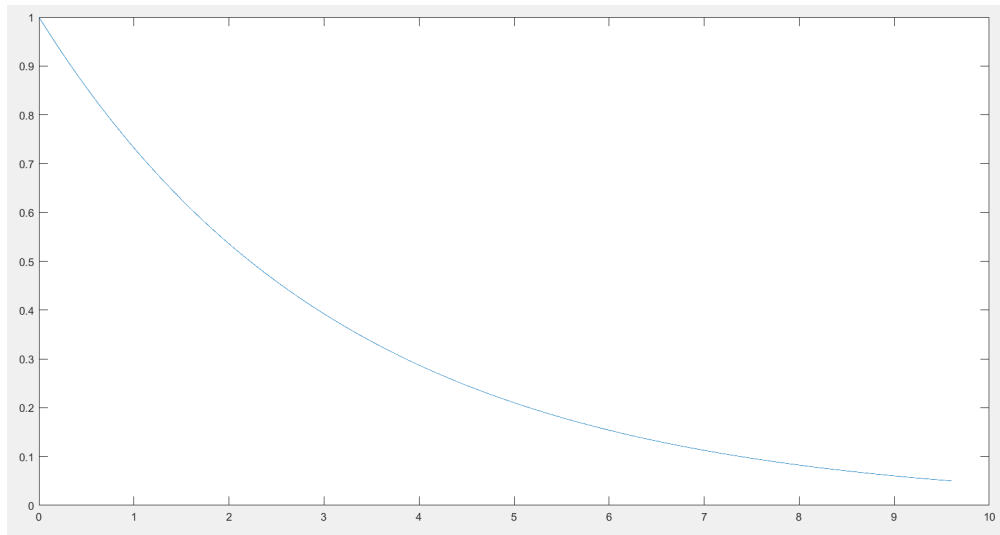


Figure 4: Negativ eksponentiel funktion

På Figure 4 kan man se, hvordan vores eksponentielle funktion skal se ud. For at gøre dette skal man først have en funktion som laver dette fald fra 100% til 5%. Til dette bruger jeg følgende formel:

$$f(x) = b \cdot a^x$$

Da vi har et fald fra 100% ved vi at b skal være 1. Dermed kan vi forkorte formelen til følgende:

$$f(x) = a^x$$

Herefter skal vi ha fundet vores a og dette kan man gøre ud fra følgende formel:

$$a = \frac{y_2}{y_1}^{\frac{1}{x_2 - x_1}}$$

Her skal vi indsætte vores tidsrum $x_2 - x_1$ og vores breddefald $\frac{y_2}{y_1}$.

Fra opgaven ved vi at tidsrummet er fra sidste tredjedel til slutningen, samt at breddefaldet går fra 100% til 5%. Så med følgende kode kan man lave en eksponentiel værdi som man bagefter vil kunne gange på den sidste tredjedel af vores samples fra signal 3.

Listing 2: Oprettelse af eksponentielt aftagende funktion

```
1 t = y(4).time(1:y(4).nS/3+1);
2 a = (0.05/1)^(1/(t(end)-t(1)));
3 e = a.^t;
```

Herefter har vi fået e som er en array på størrelsen af den sidste tredjedel af signal 3 og har procentværdierne som man kan gange på vores originale signal.

Listing 3: Oprettelse af eksponentielt aftagende funktion

```
1 y(7).sample(end-length(t):end) = e.* y(7).sample(end-length(t):end);
```

Så ved at køre koden fra Listing 3 kan man få en eksponentiel aftagende sidste tredjedel af signal 3. Dette kan ses på Figure 5.

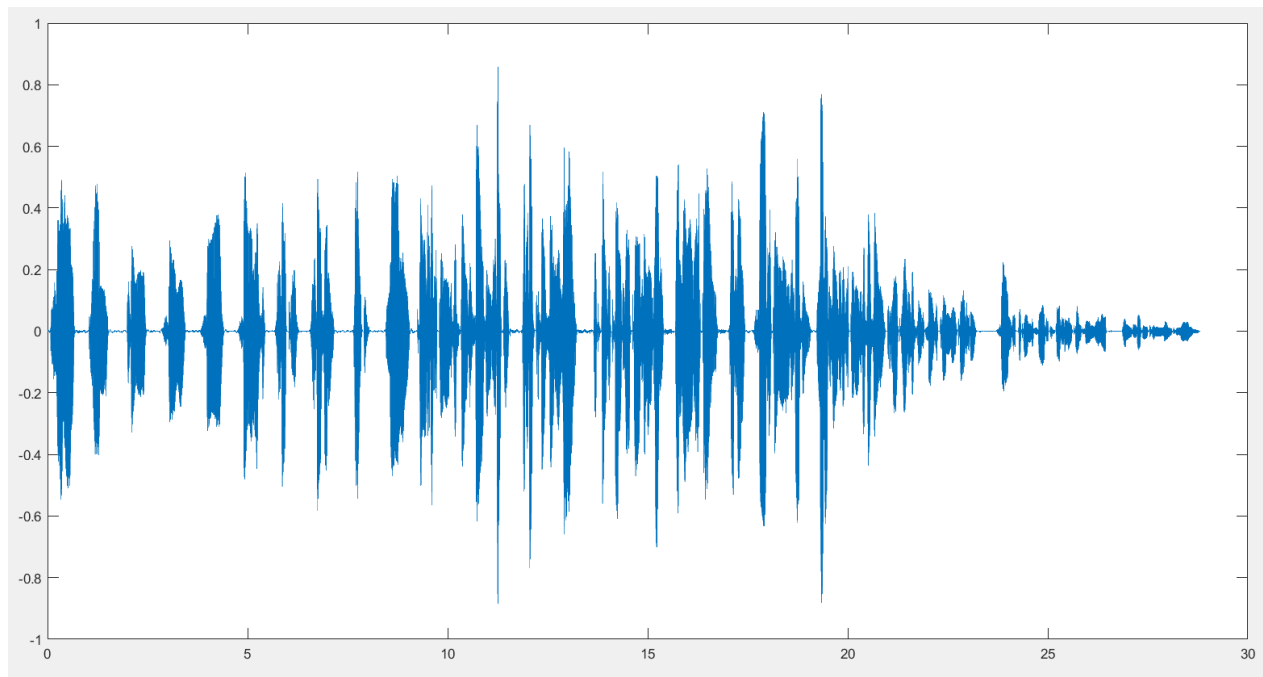


Figure 5: Fadeout på signal 3