

Opgave 9

For at få en eksponentielt aftagning på s3, kræver det at s3 ganges med en "envelope" der kan fungere som der ønskes. Her at der på den sidste 1/3 af s3, skal aftages eksponentielt, ned til en amplitude på 5% af den originale amplitude. Dette kan opnås med funktionen for en eksponentielt aftagende funktion, da vi kender vores skæring med y-aksen, og 2 punkter på grafen, skæringen med y-aksen i (0,1) samt slutpunktet, der skal være ($\frac{1}{3}time, 0.05$). Funktionen kan ses her under i ligning 1.

$$y(x) = b \cdot a^x \quad (1)$$

$$b = \text{y-værdi for skæring med y-aksen} \quad (2)$$

$$a = \left(\frac{y_2}{y_1}\right)^{\frac{1}{x_2 - x_1}}, \text{ hvor } (x_1, y_1) \text{ og } (x_2, y_2) \text{ er punkter på grafen} \quad (3)$$

Listing 1: Fade out af s3label

```
1 %% Exercise 9
2 % Tager sidste tredjedel af s3
3 t = y(s3).time(1:y(s3).nS/3+1);
4 % Eksponentielt aftagende: b*a^x
5 % b = Skæring med y i (0,1)
6 % a = (y2/y1)^1/(x2-x1)
7 a = (0.05/1)^(1/(t(end)-t(1)));
8 % Matrice der er eksponentielt aftagende
9 % Starter i 1 og går til 0.05 ved 1/3 af time
10 e = 1*a.^t;
11
12 % s3fade
13 s3fade = 7;
14 y(s3fade).sample = y(s3).sample;
15 y(s3fade).time = y(s3).time;
16 % Påfør e på sidste 1/3 af samples
17 y(s3fade).sample(length(y(s3fade).sample)*2/3:end)
18     = e.*y(s3fade).sample(length(y(s3fade).sample)*2/3:end);
19
20 figure
21 plot(y(s3fade).time, y(s3fade).sample, Fs)
22 %soundsc(y(s3fade).sample)
```

Som det kan ses på figur 1, er den ønskede effekt opnået. På den sidste 1/3 af signalet, bliver stemmen dæmpet eksponentielt. Ved at lytte til signalet kan det også høres, den ønskede effekt er opnået.

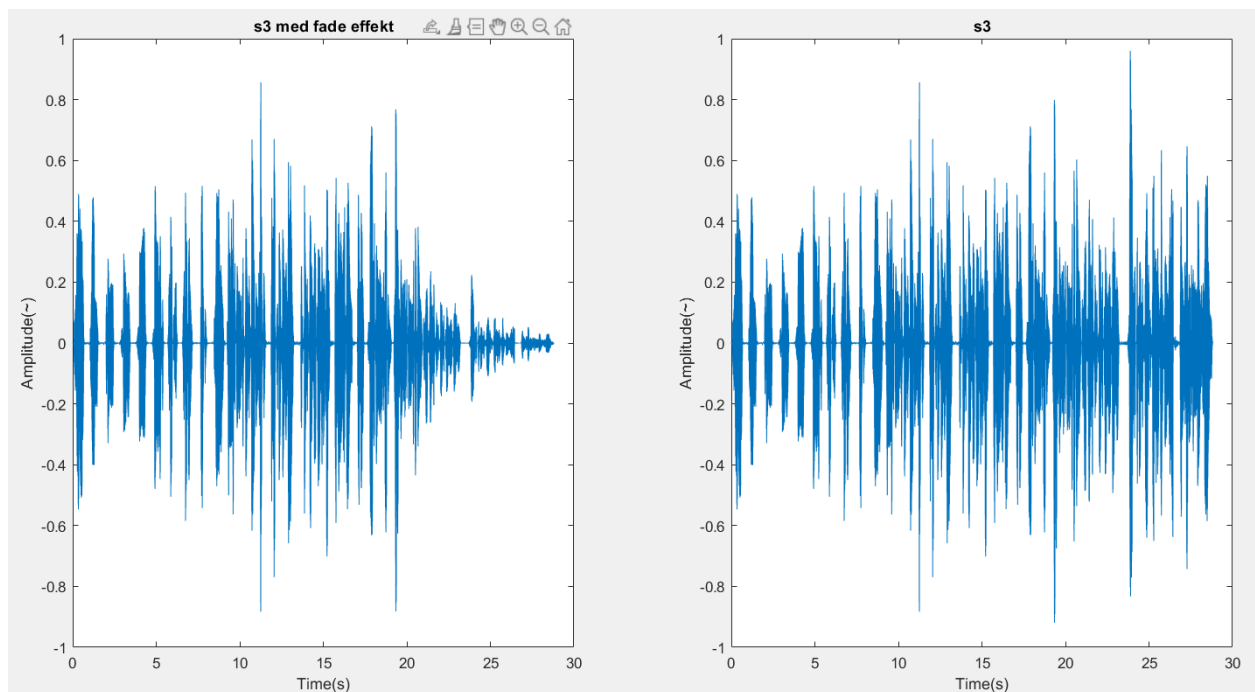


Figure 1: s3 med fade effekt

Opgave 1.15

Til denne opgave ønskes der at simulere et ”ekko” på et tone signal, hvor ekkoet har forskellige forskydning, til tonen. Først køres der med et ekko, der er 150ms forsinket, der efter et ekko med en forsinkelse på 40ms og 300ms. Tonen der er valgt her er 440Hz, kammertonen.

Listing 2: Kode til opgave 1.15

```

1 Fs = 5000; %Samplingsfrekvens
2 Ts = 1/Fs; % Tid per sampling
3
4 length = 1; % Længde i sekunder
5 Ns = Fs*length; % Antallet af samples
6 t = [0:Ns-1]*Ts; % Tiden over samplingen
7
8 %Tone beskrivelse
9 f0 = 2350; %440 Hz tone
10 A = 3; % Amplitude er 3
11 tone = A * sin(f0*2*pi*t); %Tonen som funktion
12
13 e.delay = 150; %ekko med 150 ms forsinkelse
14 e.Ns = e.delay * (Fs/1000); %Antallet af samples i ekkoet
15
16 tone_ekko = [zeros(1,e.Ns), tone]; %Forsinker ved at smide 0'er ind foran
17 tone_ext = [tone, zeros(1,e.Ns)]; % Forlænger ved at smide 0'er ind bagved

```

```

18
19 tone_sum = tone_ekko + tone_ext;
20 ekko_t = [0:Ns+e.Ns-1]*Ts;

```

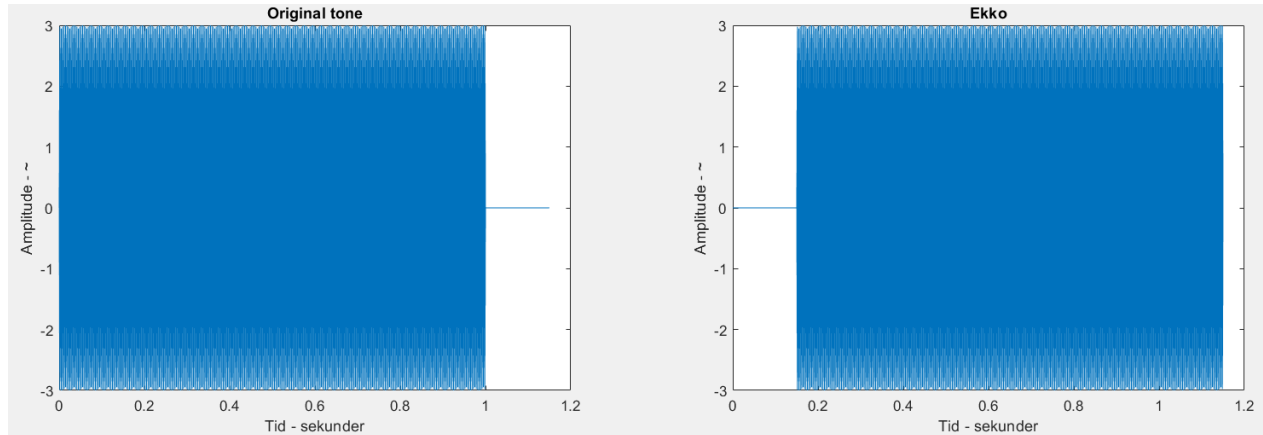


Figure 2: Plot af 440Hz tone med ekko

På figur 2 kan ses hvordan den originale tone ser ud, ved siden af sit ekko. Ekko effekten er lavet, ved at sætte et antal nuller foran det originale signal, for på den måde at "udskyde" hvornår signalet starter. Derfor er der også sat ekstra nuller bag på det originale signal, da de 2 matricer ellers ikke vil være lige store.

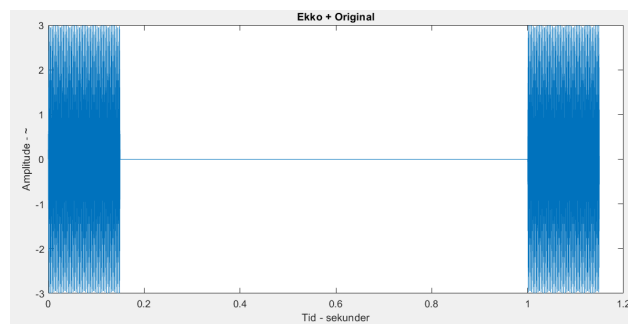


Figure 3: Plot af sum af 440Hz tone og Ekko på 150ms

På figur 3 ses hvad summen af 440Hz tonen, sammen med sit ekko ser ud. Vi kan se, der hvor de 2 toner overlapper, har de interfereret, og gået ud med hindanden. Dette skyldes forskydningen af ekko-signalet, der er blevet drejet 180 deg i forhold til det originale signal. Effekten høres tydeligt ved at afspille tonen, da der cuttes hurtigt ud, med et dødt stykke i midten, og en kort tone til sidst.

Forsøget er også blevet udført med 40ms delay og 300ms delay. I disse tilfælde har tonen ikke været forskudt, og original tonen og ekkoet har derfor i stedet forstærket hindanden. Denne effekt kan ses på figur 4. Her kan også ses at amplituden for interference signalet er fordoblet, til en amplitude på 6.

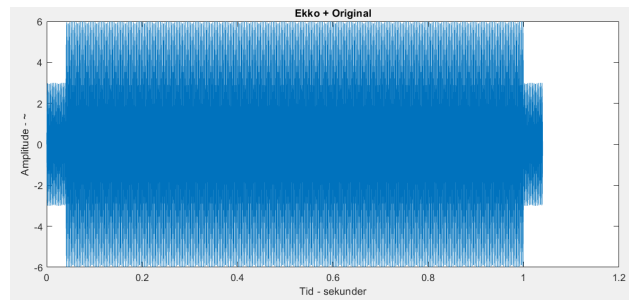


Figure 4: Plot af sum af 440Hz tone og Ekko på 40ms