

## 0.1 Introduction

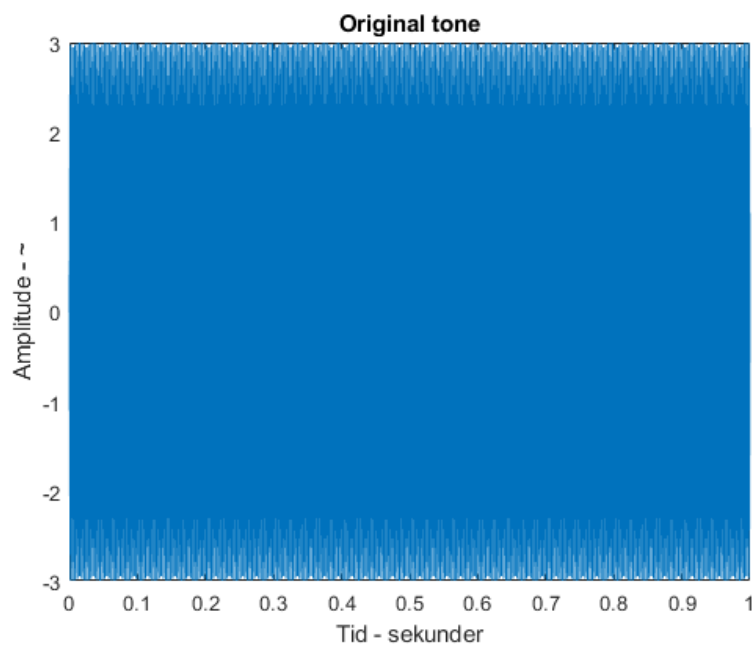
I den sidste del af øvelsen vil jeg besvare opgave 1.15 fra DSB uge 38. Opgaven går ud på at konstruere et ekko-signal på 150ms. Jeg vil efterfølgende variere ekkoet til 40ms og 300ms respektivt.

## 0.2 Fremgangsmåde

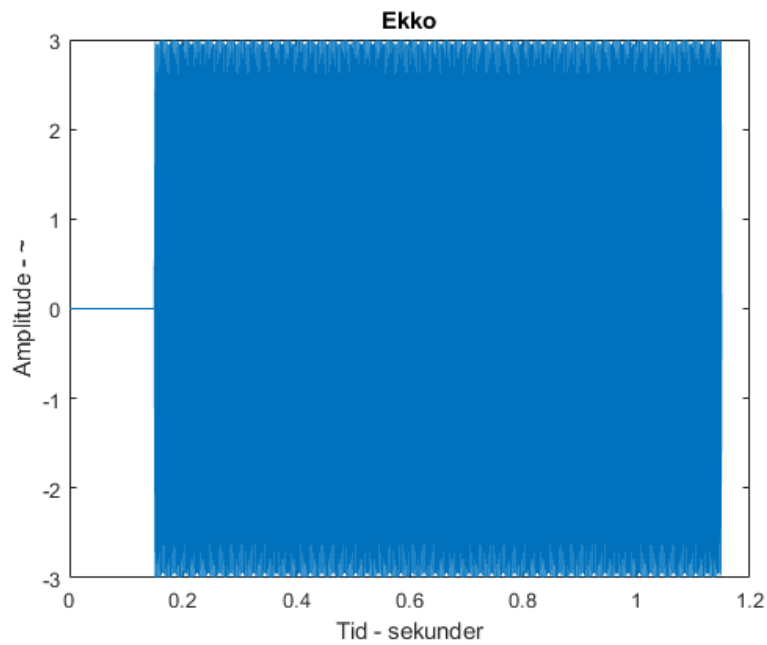
**Listing 1:** Matlab kode for øvelse 1.15, uge 39 DSB

```
1 f_s = 5000; %Samplingsfrekvens
2 T_s = 1/Fs; % Periode
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 f = 2350; %440 Hz
10 A = 3; % Amplitude er 3
11 sig = A * sin(f0*2*pi*t); %funktion for signal
12
13 ekko.delay = 150; %ekko med 150 ms forsinkelse
14 ekko.Ns = ekko.delay * (f_s/1000); %Antallet af samples i ekkoet
15
16 tone_ekko = [zeros(1,ekko.Ns), sig]; %Forsinker ved at sætte 0-array foran
17 tone_ext = [sig, zeros(1,ekko.Ns)]; % Forlænger array ved at sætte 0-array bagved
18
19 tone_sum = tone_ekko + tone_ext
20 ekko_t = [0:Ns+ekko.Ns-1]*T_s;
21
22 figure
23 plot(t, sig);
24 title("Original tone");
25 xlabel("Tid - sekunder");
26 ylabel("Amplitude - ~");
27
28 figure
29 plot(ekko_t, tone_ekko);
30 title("Ekko");
31 xlabel("Tid - sekunder");
32 ylabel("Amplitude - ~");
33
34 figure
35 plot(ekko_t, tone_sum);
36 title("Ekko + Original");
37 xlabel("Tid - sekunder");
38 ylabel("Amplitude - ~");
39
40 soundsc(tone_sum)
```

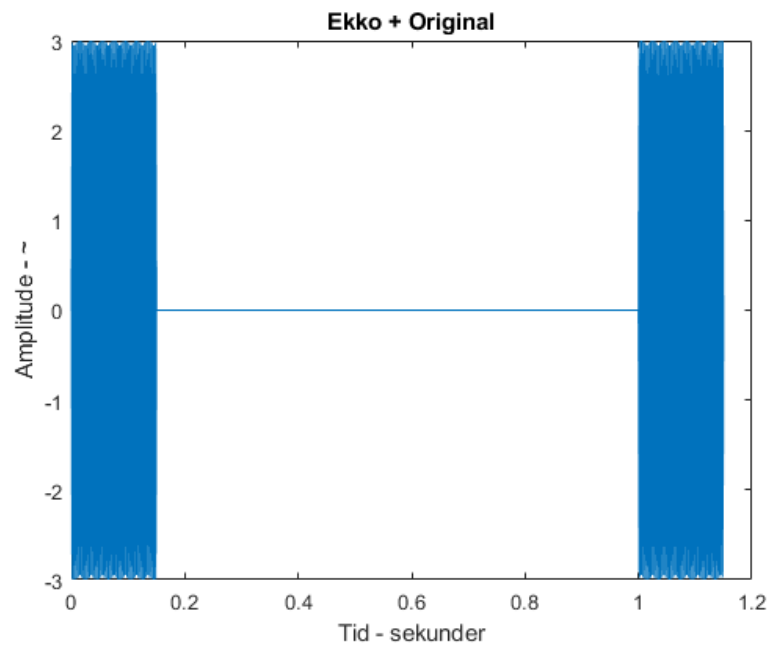
### 0.3 Resultater



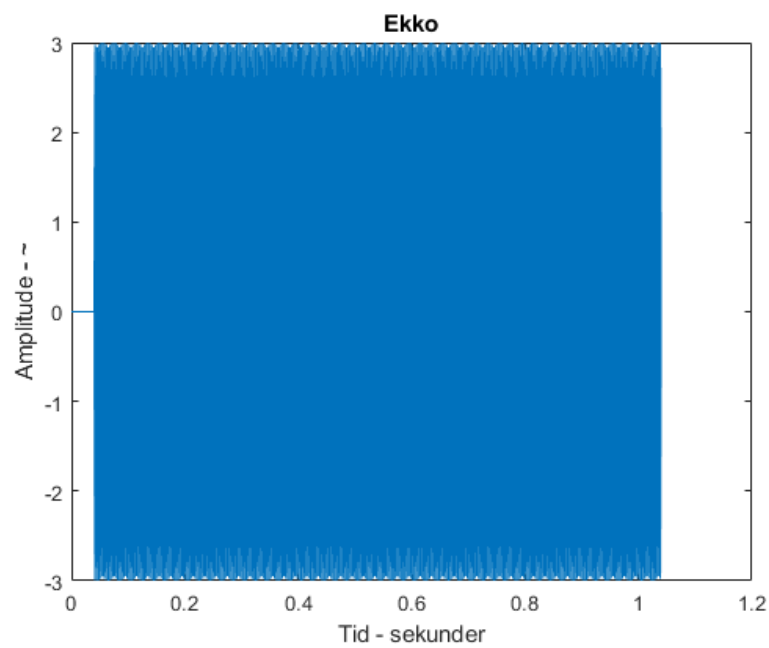
**Figure 1:** Signal *sig* uden ekko



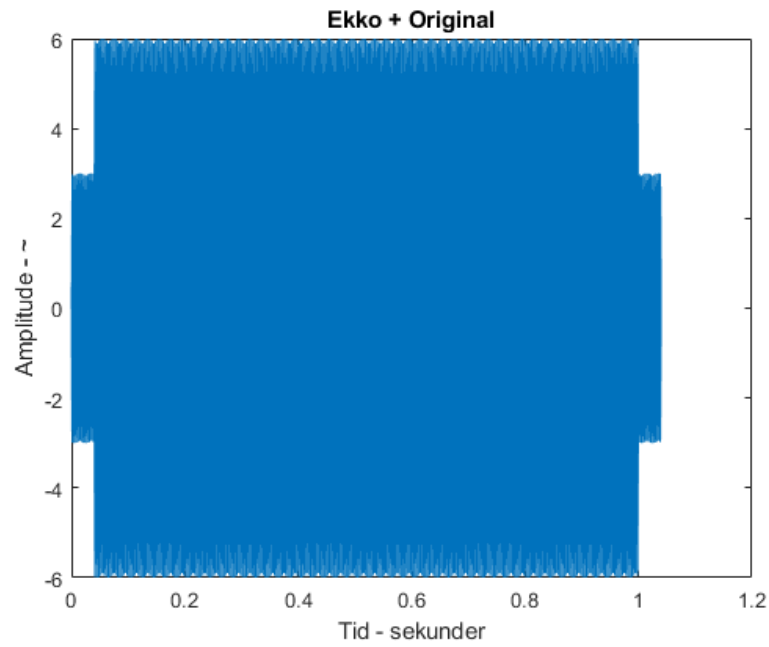
**Figure 2:** ekko signal på 150ms af *sig*



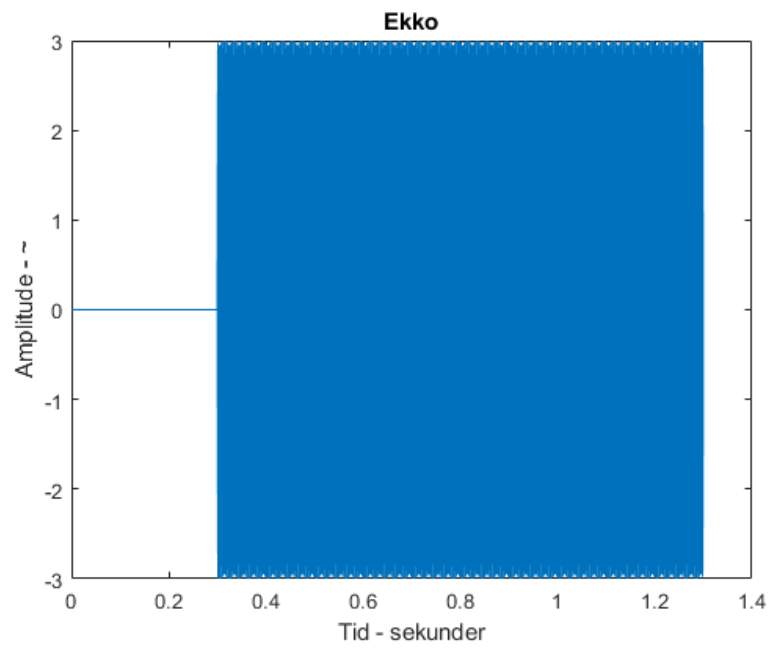
**Figure 3:** Interferens mellem de to signaler, 150ms



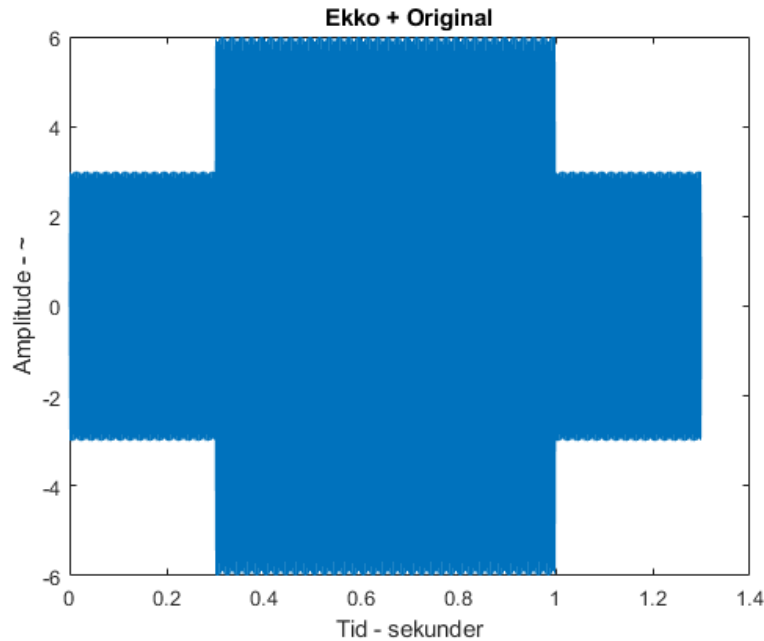
**Figure 4:** Ekko signal på 40ms af *sig*



**Figure 5:** Interferens mellem de to signaler, 40ms



**Figure 6:** Ekko signal på 300ms af *sig*



**Figure 7:** Interferens mellem de to kanaler, 300ms

## 0.4 Diskussion

For at kunne plotte signalet laver jeg først en ny tidsakse ud fra samplingfrekvensen  $f_s$  og perioden  $T_s$ . Jeg opstiller derefter min funktion for signalet,  $sig$ , og definerer derefter mit ekko-delay. Jeg benytter `zeros` funktionen til at indsætte nuller foran sig-signalet, så det får det ønskede delay. Ligeledes gør jeg det samme, dog nu med nullerne bagefter. Dette er så de har samme mængde elementer i deres arrays, og kan plottes sammen. Jeg plotter så den originale signal-funktion sammen med ekko-funktionen.

På figur 12 og 13 kan man se de to funktioner plottet, og på figur 14 ses hvordan de interferere. Det ses at der fra ca. 150ms til 1000ms forekommer stærk destruktiv interferens, og de to signaler ødelægger hinanden, idet de er faseforskudt med  $\pi$  radianer. på figur 15 og 17 ses istedet resultatet af interferens med et ekko signal på 40ms og 300ms henholdsvis. Begge steder ser jeg konstruktiv interferens, altså sker der en addition af de to signals amplituder. Altså må signalet  $sig$  være faseforskudt med 0 grader med både 40ms og 300ms ekkoet.

## 0.5 Konklusion

Jeg konkludere at man kan konstruere et ekko-signal vha. Matlab's `zeros()` funktion. Jeg har desuden set hvordan Matlab kan benyttes til at analysere interferens mellem signaler, og bl.a kan bruges til at bestemme hvornår der vil ske konstruktiv og destruktiv interferens.