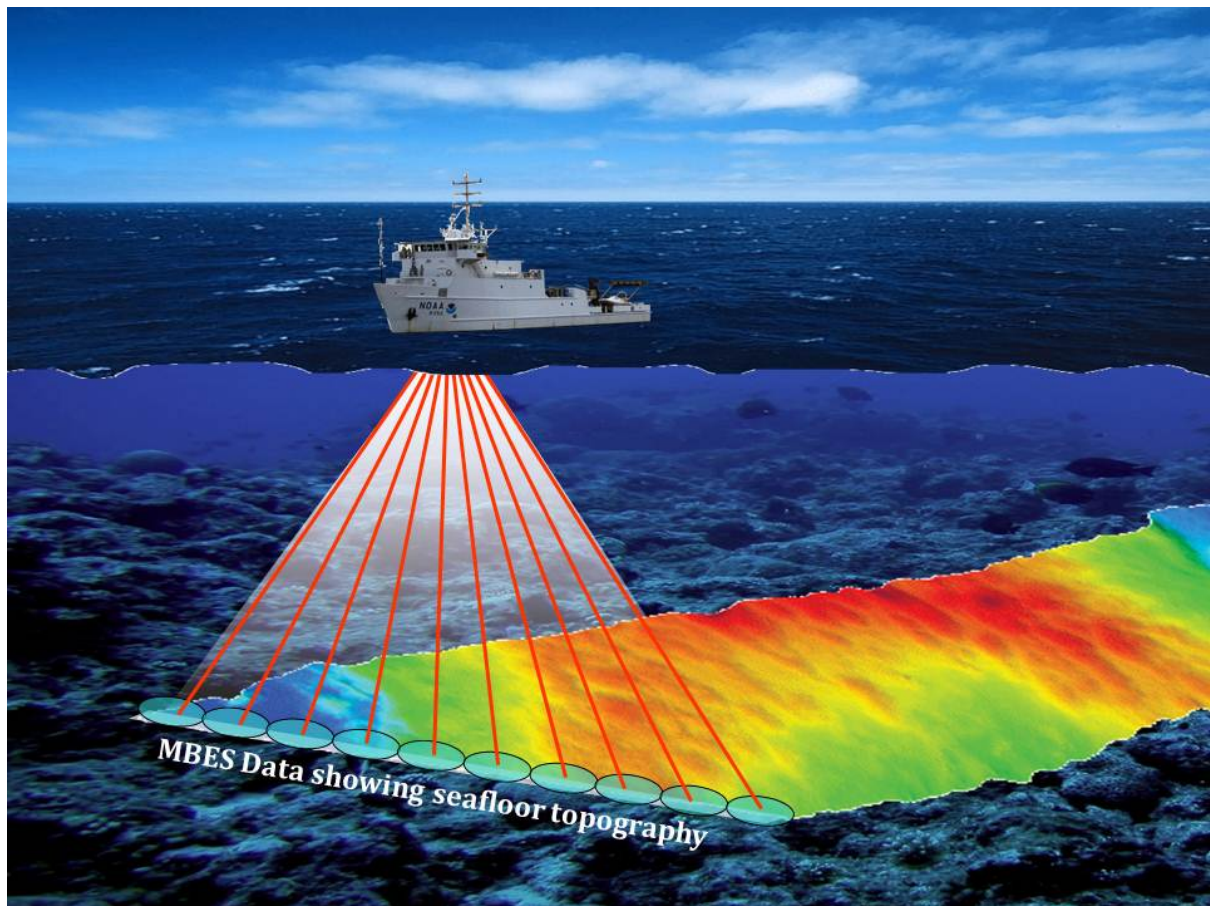


E4DSA Case 4 - Sonar

Janus Bo Andersen ¹

8. maj 2020



¹ja67494@post.au.dk

Indhold

1	Indledning	1
2	Opgave 1: Signalgenerering og simulering	1
2.1	Teori	2
2.1.1	Krydskorrelation	2
2.1.2	Tidsforskydning, minimumsafstand og afstandsopløsning	2
2.1.3	Indflydelse fra støj	3
2.2	Valg af afsendersignal	3
2.3	Simulering af sonar	5
2.3.1	Simulering med mere støj	8
2.3.2	Simulering med lavpasfiltrering	9
3	Konklusion	12
4	Kildehenvisning	14
5	Funktioner	15
5.1	setlatexstuff	15

1. Indledning

Fjerde case i E4DSA omhandler sonar (SOund Navigation Ranging). Ideen er at udsende et eller flere pulsede signaler, der reflekteres på et target. De reflekterede signaler (ekkoer) opfanges og analyseres for at kunne bestemme fx targets (1) afstand, (2) hastighed (Doppler-effekt), (3) bevægelsesretning eller (4) form/struktur/størrelse (signatur).

Sonar er nok ensbetydende med anvendelser *under* vand og med signaler båret i lydbølger. Det kendes f.eks. fra u-både <https://www.youtube.com/watch?v=jr0JaXfKj68>. Her kan sonar bruges til at kortlægge eller navigere i havbunden - eller til at lokalisere skibe og andre objekter i vandet. På forsiden ses et eksempel af en Multibeam echosounder (MBES), der kan kortlægge havbundens topografi vha. sonar-“stråler” (vha. beamforming). Biosonar (ekkolokalisering), som hvaler og delfiner anvender til kommunikation og jagt, er et interessant eksempel fra naturen.

Over vandet anvendes ideen fx i ultrasoniske sensorer i biler. Fx som sensorer til at afstandbedømme og assistere ved parkering og bakning. Det er et billigt alternativ til LIDAR. Der sidder også en ultrasonisk sensor i bunden af min quadrotor-drone, som (sammen med et barometer) benyttes til højdemåling. I disse anvendelser benyttes luft som bæremedium, der naturligvis har helt anderledes akustiske karakteristika end vand.

Radar-systemer er et parallelt eksempel, hvor signalet i stedet bæres af elektromagnetiske bølger og med en meget højere udbredelseshastighed. Transducere er også væsentlig mere komplicerede.

Summa summarum, der er rigtig mange interessante anvendelser og perspektiver :-)

2. Opgave 1: Signalgenerering og simulering

Opgaven minder en del om Case 1 (FSK-transmission). Men i denne case ønsker vi at lave detektion af et tidsforskydet signal i *tidsdomænet* i stedet for flere forskellige frekvenser i frekvensdomænet. Så vi benytter krydskorrelation i stedet for foldning¹. I det følgende benyttes notationen samplingsfrekvens f_s og samplingstid $T_s = 1/f_s$.

¹ Hvilket så er ligesom foldning, hvis man pre-flipper det ene af signalerne

2.1 Teori

2.1.1 Krydskorrelation

Krydskorrelation er givet ved [1, kap. 7]

$$c(n) = \sum_{k=0}^{N_p-1} p(k) \cdot r(n+k) \quad (2.1)$$

Her er $p(n)$ den afsendte puls med længden N_p samples, så tidslængde af p er $N_p \cdot T_s$ [s]. Signalet $p(n)$ forsøges detekteret i det reflekterede signal $r(n)$. Krydskorrelation til tidspunkt n er som et indre produkt af p og et udvalgt interval af $r(n)$, $n \dots n + N_p$. Med tilpas zero-padding, hvis nødvendigt.

$$c(n) = \langle p, r_{n:n+N_p} \rangle(n) \quad (2.2)$$

Hvis signalerne er ortogonale (helt uens) i det valgte interval er korrelationen 0. En numerisk høj krydskorrelation angiver høj energi af signalet p ved $r(n)$. Et højt negativt tal svarer til modfase. Detektionen kan afgøres ved at sætte en tærskel-værdi.

Rullende krydskorrelation giver en tidserie, der viser ved hvilke tidsforskydninger p kan detekteres i $r(n)$. Her er det ønskeligt at have så smalt et "blip" i krydskorrelationen som muligt, for at få en maksimalt præcis afstandsmåling.

2.1.2 Tidsforskydning, minimumsafstand og afstandsopløsning

Tidsforskydning mellem afsendt og modtaget puls er en funktion af afstand til target, R [m], og signalets hastighed, v [m/s] [2]. Tidsforskydningen er τ [s], svarende i samples til $\tau = n_\tau \cdot T_s$.

$$R = \frac{v \cdot \tau}{2} \iff \tau = \frac{2R}{v} \quad (2.3)$$

Med sampling (diskret tid)

$$R = \frac{v \cdot n_\tau \cdot T_s}{2} = \frac{v \cdot n_\tau}{2f_s} \iff n_\tau = \frac{2R}{T_s \cdot v} = \frac{2R}{v} f_s \quad (2.4)$$

For vores sonar er τ tiden det tager signalet at tilbagelægge rundturen ud til target og tilbage i fri luft. En lydbølges udbredelseshastighed i fri luft ved 20 °C er $v = 343$ m/s.

Så hvis der i vores system observeres tidsforskydning på $n_\tau = 1$ sample, fx med samplingsfrekvens 1 kHz, så er afstanden til target på $R = \frac{343 \cdot 1}{2 \cdot 1000} = 0.172$ [m].

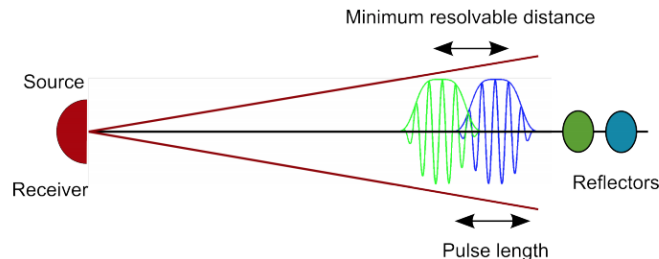
Minimumsafstand: Hvis afsendelse af signal skal være afsluttet inden ekko returnerer, skal følgende gælde for en en puls med bredden T_{puls} [s]:

$$T_{\text{puls}} < \tau = \frac{2R}{v} \quad (2.5)$$

Det betyder, at hvis et objekt er på afstanden 1.00 m, skal $T_{\text{puls}} < \frac{2 \cdot 1.00 \text{ m}}{343 \text{ m/s}} = 0.00583 \text{ s}$. Altså en puls på maks. 5.83 ms.

For en ren sinustone kan en konstant lyttende modtager ved siden af senderen ikke adskille det afsendte signal fra et ekko. Det er fordi $p(n)$ korrelerer periodisk med sig selv (autokorrelation) og med ekko (krydskorrelation). Så her er $T_{\text{puls}} < \tau$ vigtigt! For et ikke-periodisk signal er relationen mindre væsentlig, antaget at der ikke sker destruktiv interferens.

Afstandsopløsning: Opløsningen bestemmes af pulsbredden. Når to returpulser overlapper, kan to targets ikke længere adskilles. Dette er illustreret i figuren nedenfor [2, s. 34].



Figur 2.1: Opløsning for afstandsmåling

Så opløsningen på afstandsmåling, ΔR , er

$$\Delta R = \frac{v \cdot T_{\text{puls}}}{2} \quad (2.6)$$

Med en varighed på fx $T_{\text{puls}} = 0.1$ s er opløsningen $\Delta R = \frac{343\text{m/s} \cdot 0.1\text{s}}{2} = 17.2$ m. Med en frekvens i den høje ende af det hørbare område, fx $f_0 = 20$ kHz, kan man afsende $f_0 \cdot t = 100$ perioder på $t = 5.0$ ms og opnå en opløsning på 0.86 m. Det kræver så en transducer, der kan afsende og modtage dette (måske et piezo-element) samt ADC der kan klare $f_s > 40$ kHz.

2.1.3 Indflydelse fra støj

Antag der afsendes en impuls med amplitude A , så $p(n) = A\delta(n)$. Så modtages der tidsforskuet en impuls, som pga. afstand og medium er dæmpet med faktor K samt tillagt additiv støj. Lydtryk for en lydbølge falder af med $K = \frac{1}{R}$. Da bliver det modtagne signal $r(n) = AR^{-1}\delta(n - n_\tau) + B(n)$. Krydskorrelationen bliver

$$c(n) = \sum_{k=0}^{N_p-1} A\delta(k) \cdot AR^{-1}\delta(n - n_\tau + k) + \sum_{k=0}^{N_p-1} A\delta(k) \cdot B(n+k) = \begin{cases} \frac{A^2}{R} + A \cdot B(n), & \text{hvis } n = n_\tau \\ A \cdot B(n), & \text{hvis } n \neq n_\tau \end{cases} \quad (2.7)$$

Så det afhænger af $\frac{A^2}{R}$, dvs. effekt af afsendt signal over afstand, set i forhold støjens størrelse, om signalet kan detekteres. Der skal altså afsendes med større effekt for at detektere på større afstand. Tilsvarende hvis støjen øges.

En ren impuls kan selvfølgelig ikke afsendes over en analog kanal. Det ville heller ikke være en god idé. Så der skal bruges en anden signaltype.

2.2 Valg af afsendersignal

Overblik over typer og anvendelser:

- Hvis signalet skal bruges til *hastighedsmåling* med Doppler-effekten, så skal frekvensen være konstant.
- Hvis signalet skal bruges til præcis *afstandsmåling*, er det vigtigt at “spike” i krydskorrelationen er meget smal, hvilket kræver en (frekvens)modulation.
- Hvis signalet skal begge dele, så kan det nok komponeres (kodes), så der både er elementer til Doppler-beregning og til afstandsbedømmelse.

Tre oplagte muligheder til at implementere afsendersignal er derfor pulser bestående af:

- Rene sinustoner (\rightarrow Dopplerberegning). *MEN*: $p(n)$ krydskorrelerer med sit ekko for hver periode i sinussignalet, så det giver et bredt “blip”.
- Chirps, frekvensmoduleret sinus (\rightarrow afstandsmåling). Korrelerer kun væsentligt når “forkanterne” på signal og ekko flugter. Det giver et smalt og kraftigt “blip”. *MEN*: frekvensen er moduleret, så Doppler-beregning er vanskelig.
- Et kodet signal (fx FSK, dog uden for scope i denne case). Kunne gøre begge dele.

Trade-off på længden af pulsen:

- Lang nok til at få en detekterbar og ikke falsk-positiv effekt i krydskorrelationen (støjimmunitet).
- Tilpas kort, så der fås en høj opløsning (lav minimumsafstand).

Trade-off på frekvensen i pulsen:

- Højere frekvenser \rightarrow mange perioder sendes afsted på kort tid: Flere datapunkter til krydskorrelation, bedre afstandsopløsning ΔR . *MEN*: luften har tendens til at lavpasfiltrere, så højere frekvenser vil opleve større dæmpning.
- Lavere frekvenser: Dæmpes ikke nær så meget i luften, så kan opnå større afstande. *MEN*: Behov for at sende længere puls \rightarrow dårligere frekvensopløsning og minimumsafstand.

Til afstandsmåling i denne case er **chirp-signalet** mest velegnet. Der vælges et setup som nedenfor.

```
1 clc; clear all; close all;
2 rng(0); % seed så vi får samme resultater hver gang
3 setlatexstuff('latex');
```

```
1 fs = 44100; % Samplingsfrekvens [Hz]
2 Tp = 0.1; % Pulsbredde [s]
3 v = 343; % Lydbølges udbredelseshastighed i luft [m/s]
4
5 dR = v * Tp / 2; % Opløsning for afstandsmåling
6 disp(['Opløsning til afstandsmåling: ' num2str(dR) ' m.'])
```

Opløsning til afstandsmåling: 17.15 m.

2.3 Simulering af sonar

I dette afsnit simuleres en chirp-puls. Den sammenlignes med en ren sinus, for at se forskelle i krydskorrelationen og få et indblik i støjimmuniteten. Chirp-pulsen er en lineært frekvensmoduleret puls, der genereres som beskrevet i [3, s. 414]. Frekvensen moduleres op fra 0 Hz til 500 Hz over pulsbredden. For den rene sinus benyttes 500 Hz til hele pulsen.

```
1 A = 10; % Amplitude for puls
2 F1 = 500; % Signalfrekvens [Hz] (slutfrekvens for chirp)
3 L = Tp * fs; % Pulslængde (4410 samples)
4 n = (0:L-1); % Samplevektor (0..4409)
5
6 nsq = n.^2;
7 p_chirp = A*sin(pi* (F1/fs) * (nsq/L) ); % Lineær FM, (Manolakis, s. 414)
8
9 p_sin = A*sin(2*pi*F1/fs*n); % Ren sinus
10
11 t_puls = (0:L-1)/fs; % Tilhørende tidsakse
```

Modtagne signaler, hvori ekko skal spores, er 0.5 s lange. Ekkoet optræder ved $\tau = 0.25$ s, ca. $R = 43$ m. Der er additiv gaussisk støj med $\sigma_r = 0.1$ (øges senere).

```
1 Tr = 0.5; % Længde af modtaget signal [s]
2 Nr = Tr*fs; % Antal samples i modtaget signal
3
4 tau = 0.25; % Tidsforskydning [s]
5 n_tau = tau*fs; % Tilsvarende sample starter ekko (11025)
6
7 R = v*tau/2; % Simuleret afstand [m] (ca. 43 m)
8 sigma_r = 0.1; % Std.afv. for støj i optaget signal
9
10 rn = randn([1 Nr]); % Gaussisk støj, genbruges senere
11 r = sigma_r*rn; % Gaussisk støj med ønsket std.afv.
12 tr = (0:Nr-1)/fs; % Tilhørende tidsakse
```

Begge signaler benytter samme samplede støjvektor. De dæmpede ekkopulser lægges ind svarende til en tidsforskydning på $\tau = 0.25$ s. Her inkluderes også dæmpning med $K = R^{-1}$ pga. afstand.

```
1 r_chirp = r; % Optaget signal med chirp-puls
2 r_chirp(1,n_tau:n_tau+L-1) = r_chirp(1,n_tau:n_tau+L-1) + p_chirp/R;
3
4 r_sin = r; % Optaget signal med sinus-puls
5 r_sin(1,n_tau:n_tau+L-1) = r_sin(1,n_tau:n_tau+L-1) + p_sin/R;
```

Krydskorrelationen beregnes med indre produkter, som beskrevet i teori-afsnittet. MATLABs `xcorr`-funktion kunne også benyttes, denne metode giver bare mere føling med beregningen.

```

1 % Tomme vektorer til resultater af rullende krydskorrelation
2 c_chirp = zeros([1 length(r_chirp)-length(p_chirp)]);
3 c_sin = zeros([1 length(r_sin)-length(p_sin)]);
4
5 % Lav krydskorrelation for både chirp og sinus
6 for n=1:length(c_chirp)
7     c_chirp(n) = p_chirp*r_chirp(n:n+L-1)';    % Regn indre produkter
8     c_sin(n) = p_sin*r_sin(n:n+L-1)';
9 end
10
11 tc = (0:length(c_chirp)-1)/fs;    % Tilhørende tidsakse

```

Grafisk sammenligning foretages nedenfor.

```

1 figure();
2 sgtitle(['Simulering af sonar' newline ...
3     '$A=$' num2str(A) ', $R=$' num2str(round(R)) ' [m]' ...
4     ', $\sigma_r=$' num2str(sigma_r) '$'], ...
5     'Interpreter', 'Latex', 'FontSize', 14);
6 subplot(321)
7 plot(t_puls, p_chirp); grid on;
8 title('Chirp-puls', 'FontSize', 12);
9 xlabel('Tid [s]', 'FontSize', 12);
10 ylabel('Amplitude', 'FontSize', 12);
11
12 subplot(322)
13 plot(t_puls, p_sin); grid on;
14 title('Sinus-puls', 'FontSize', 12);
15 xlabel('Tid [s]', 'FontSize', 12);
16 ylabel('Amplitude', 'FontSize', 12);
17
18 subplot(323)
19 plot(tr, r_chirp); grid on;
20 title('Modtaget signal inkl. chirp-puls og gaussisk', 'FontSize', 12);
21 xlabel('Tid [s]', 'FontSize', 12);
22 ylabel('Amplitude', 'FontSize', 12);
23 xlim([0 Tr])
24
25 subplot(324)
26 plot(tr, r_sin); grid on;
27 title('Modtaget signal inkl. sinus-puls og gaussisk', 'FontSize', 12);
28 xlabel('Tid [s]', 'FontSize', 12);
29 ylabel('Amplitude', 'FontSize', 12);
30 xlim([0 Tr])
31
32 subplot(325)

```

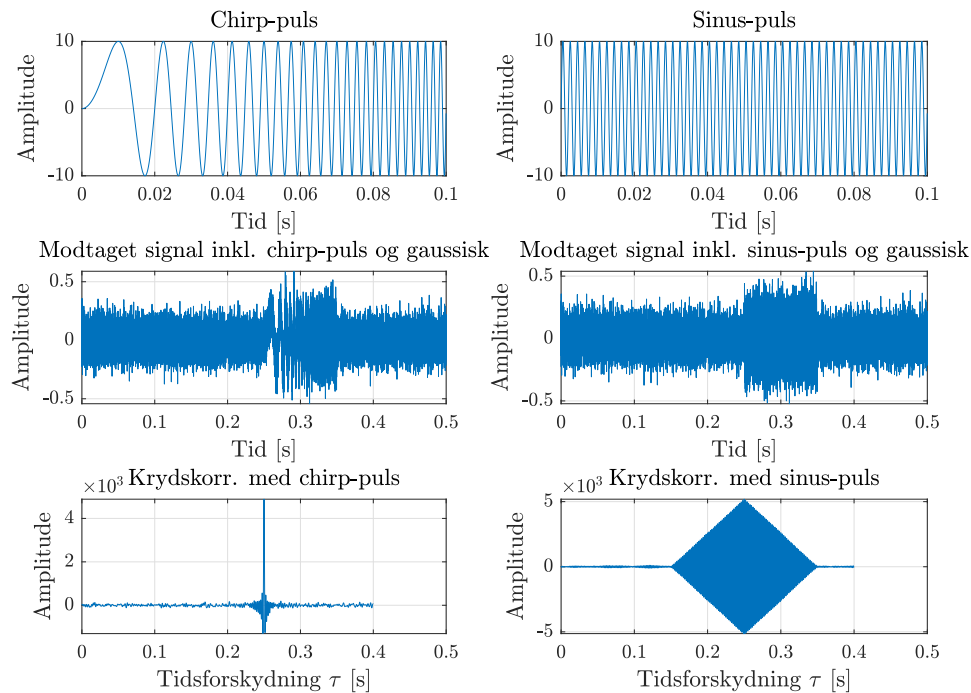


```

33 plot(tc, c_chirp); grid on;
34 title('Krydskorr. med chirp-puls', 'FontSize', 12);
35 xlabel('Tidsforskydning $\tau$ [s]', 'FontSize', 12);
36 ylabel('Amplitude', 'FontSize', 12);
37 xlim([0 Tr])
38 ax = gca; ax.YAxis.Exponent = 3;
39
40 subplot(326)
41 plot(tc, c_sin); grid on;
42 title('Krydskorr. med sinus-puls', 'FontSize', 12);
43 xlabel('Tidsforskydning $\tau$ [s]', 'FontSize', 12);
44 ylabel('Amplitude', 'FontSize', 12);
45 xlim([0 Tr])
46 ax = gca; ax.YAxis.Exponent = 3;

```

Simulering af sonar
 $A=10$, $R=43$ [m], $\sigma_r = 0.1$



Figurerne viser, som forventet, at maksimal effekt i krydskorrelationen optræder ved $\tau = 0.25$ s ($R = 43$ m). Den grafiske sammenligning dokumenterer fordelene ved chirp-pulsen (og andre frekvensmodulerede pulser): Der sker *puls-kompression* i krydskorrelationen, og al energien afsættes over meget kort tid. Det skyldes responset af den frekvensmodulerede puls i krydskorrelationen (et matched filter). Se fx [4] eller [5]. Det er, som beskrevet tidligere, fordi krydskorrelationen kun slår ud, når puls og ekko “aligner” perfekt.

Der er aliasering i figurerne med krydskorrelation. Når der zoomes ind, kan det ses, at:

- Krydskorrelation af chirp med lagget chirp giver en sinc-funktion.
- Krydskorrelation af sinus med lagget sinus giver noget, der minder om en sinus med trekantet indhyldning.

2.3.1 Simulering med mere støj

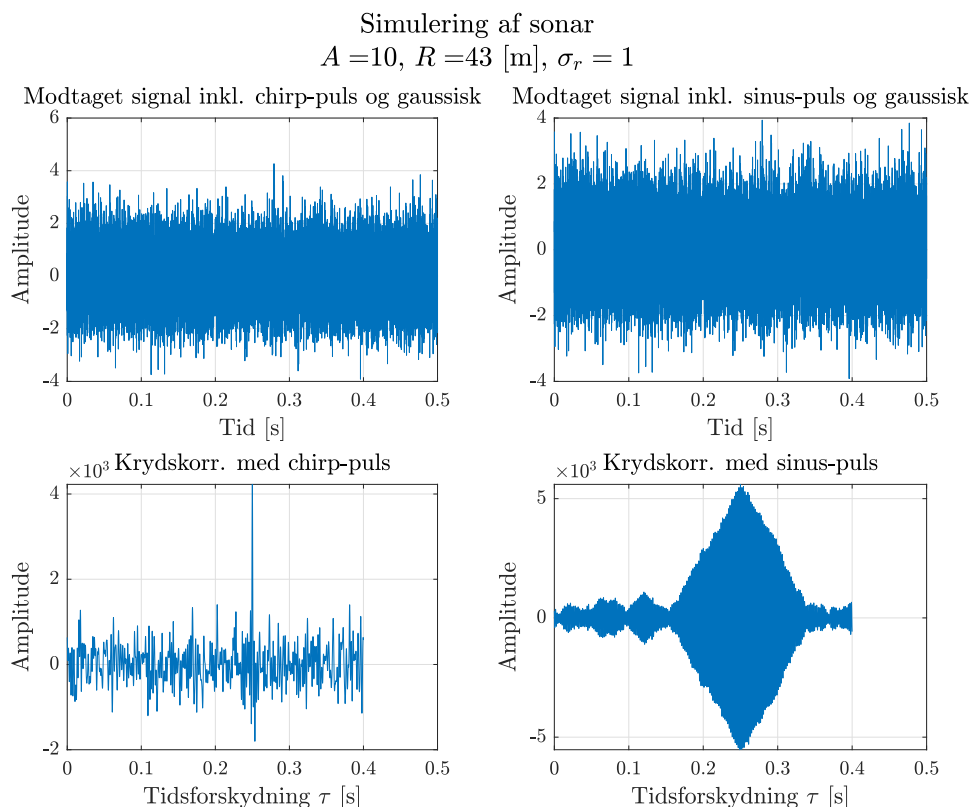
Fastholdes effekten fra afsenderen ($A = 10$) men med med faktor 10 højere støjniveau, $\sigma_r = 1$, fås nedenstående.

```
1 sigma_r = 1.0;           % Std.afv. for støj i optaget signal
2 r = sigma_r*rn;          % Genbrug samme støjvektor til sammenligning, std.afv.
3
4 % Genberegn modtagede signaler
5 r_chirp = r;              % Optaget signal med chirp-puls
6 r_chirp(1,n_tau:n_tau+L-1) = r_chirp(1,n_tau:n_tau+L-1) + p_chirp/R;
7 r_sin = r;                % Optaget signal med sinus-puls
8 r_sin(1,n_tau:n_tau+L-1) = r_sin(1,n_tau:n_tau+L-1) + p_sin/R;
9
10 % Genberegn krydskorrelation
11 for n=1:length(c_chirp)
12     c_chirp(n) = p_chirp*r_chirp(n:n+L-1)';      % Regn indre produkter
13     c_sin(n) = p_sin*r_sin(n:n+L-1)';
14 end
15
16 % Plot
17 figure();
18 sgtitle(['Simulering af sonar' newline ...
19     '$A=$' num2str(A) ', $R=$' num2str(round(R)) ' [m]' ...
20     ', $\sigma_r=$' num2str(sigma_r) '$'], ...
21     'Interpreter', 'Latex', 'FontSize', 14);
22
23 subplot(221)
24 plot(tr, r_chirp); grid on;
25 title('Modtaget signal inkl. chirp-puls og gaussisk', 'FontSize', 12);
26 xlabel('Tid [s]', 'FontSize', 12);
27 ylabel('Amplitude', 'FontSize', 12);
28 xlim([0 Tr])
29
30 subplot(222)
31 plot(tr, r_sin); grid on;
32 title('Modtaget signal inkl. sinus-puls og gaussisk', 'FontSize', 12);
33 xlabel('Tid [s]', 'FontSize', 12);
34 ylabel('Amplitude', 'FontSize', 12);
35 xlim([0 Tr])
36
37 subplot(223)
38 plot(tc, c_chirp); grid on;
39 title('Krydskorr. med chirp-puls', 'FontSize', 12);
40 xlabel('Tidsforskydning $\tau$ [s]', 'FontSize', 12);
41 ylabel('Amplitude', 'FontSize', 12);
42 xlim([0 Tr])
```

```

43 ax = gca; ax.YAxis.Exponent = 3;
44
45 subplot(224)
46 plot(tc, c_sin); grid on;
47 title('Krydskorr. med sinus-puls', 'FontSize', 12);
48 xlabel('Tidsforskydning  $\tau$  [s]', 'FontSize', 12);
49 ylabel('Amplitude', 'FontSize', 12);
50 xlim([0 Tr])
51 ax = gca; ax.YAxis.Exponent = 3;

```



Figuren viser, at signalet nu indeholder væsentlig støj: I tidsdomænet kan pulsen ikke længere direkte ses for støj. Med pulskompression kan afstanden bedømmes præcist selv med væsentlig støj. For sinus-pulsen er billedet noget mere “mudret”.

2.3.2 Simulering med lavpasfiltrering

Som tidligere nævnt, giver en højere frekvens i pulserne generelt bedre performance for “sonar”-systemet. Så sidste del af simuleringen er at indarbejde effekt fra luftens LP-filtrering af “sonar”-pulserne.

Ideen er at få et blik på hvor højt man kan presse puls-frekvensen for de to forskellige sonar-typer givet usikkerhed på luftens knækfrekvens. Den kunne jo fx ændre sig med temperatur, luftfugtighed, barometrisk tryk, ved forskellige typer nedbør, osv.

Antag, at knækfrekvensen i luften ligger omkring F_1 men med 10 pct. usikkerhed. LP-filteret får så en worst-case knækfrekvens på $0.9F_1$.

Her benyttes MATLABs `fir1`-funktion til at designe et filter med vinduesfunktion. Der er valgt et FIR-filter for at undgå ikke-lineær fase (har “atmosfærens” LP-filter ikke-lineær fase?).

```

1  fc = F1*0.9;
2  b_luft = fir1(750, 2*fc/fs);          % LP FIR-filter med 450 Hz knæk
3
4  % Lavpas-filtrér ekko-pulserne
5  p_chirp_lp = filter(b_luft,1,p_chirp);
6  p_sin_lp = filter(b_luft,1,p_sin);
7
8  % Genberegning modtagede signaler
9  r_chirp_lp = r;                      % Samme std.afv. som før, chirp
10 r_chirp_lp(1,n_tau:n_tau+L-1) = ...
11     r_chirp_lp(1,n_tau:n_tau+L-1) + p_chirp_lp/R;
12
13 r_sin_lp = r;                        % Samme std.afv. som før, sinus
14 r_sin_lp(1,n_tau:n_tau+L-1) = r_sin_lp(1,n_tau:n_tau+L-1) + p_sin_lp/R;
15
16 % Genberegning krydskorrelation
17 for n=1:length(c_chirp)
18     c_chirp(n) = p_chirp*r_chirp_lp(n:n+L-1)';    % Regn indre produkter
19     c_sin(n) = p_sin*r_sin_lp(n:n+L-1)';
20 end
21
22 % Plot
23 figure();
24 sgtitle(['Simulering af sonar' newline ...
25     '$A=$' num2str(A) ' ', '$R=$' num2str(round(R)) ' [m]' ...
26     ', $f_c=$' num2str(fc) ' [Hz]' ...
27     ', $\sigma_r=$' num2str(sigma_r) '$'], ...
28     'Interpreter', 'Latex', 'FontSize', 14);
29
30 subplot(321)
31 plot(t_puls, p_chirp_lp); grid on;
32 title('LP-filtreret chirp-puls', 'FontSize', 12);
33 xlabel('Tid [s]', 'FontSize', 12);
34 ylabel('Amplitude', 'FontSize', 12);
35
36 subplot(322)
37 plot(t_puls, p_sin_lp); grid on;
38 title('LP-filtreret sinus-puls', 'FontSize', 12);
39 xlabel('Tid [s]', 'FontSize', 12);
40 ylabel('Amplitude', 'FontSize', 12);
41
42 subplot(323)

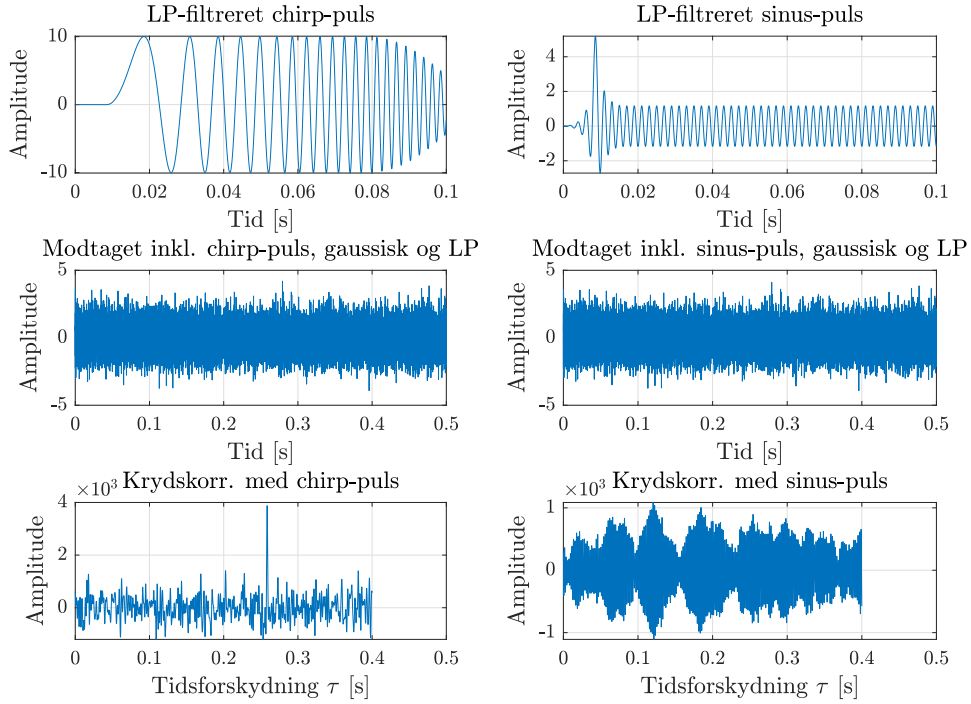
```

```

43 plot(tr, r_chirp_lp); grid on;
44 title('Modtaget inkl. chirp-puls, gaussisk og LP', 'FontSize', 12);
45 xlabel('Tid [s]', 'FontSize', 12);
46 ylabel('Amplitude', 'FontSize', 12);
47 xlim([0 Tr])
48
49 subplot(324)
50 plot(tr, r_sin_lp); grid on;
51 title('Modtaget inkl. sinus-puls, gaussisk og LP', 'FontSize', 12);
52 xlabel('Tid [s]', 'FontSize', 12);
53 ylabel('Amplitude', 'FontSize', 12);
54 xlim([0 Tr])
55
56 subplot(325)
57 plot(tc, c_chirp); grid on;
58 title('Krydskorr. med chirp-puls', 'FontSize', 12);
59 xlabel('Tidsforskydning  $\tau$  [s]', 'FontSize', 12);
60 ylabel('Amplitude', 'FontSize', 12);
61 xlim([0 Tr])
62 ax = gca; ax.YAxis.Exponent = 3;
63
64 subplot(326)
65 plot(tc, c_sin); grid on;
66 title('Krydskorr. med sinus-puls', 'FontSize', 12);
67 xlabel('Tidsforskydning  $\tau$  [s]', 'FontSize', 12);
68 ylabel('Amplitude', 'FontSize', 12);
69 xlim([0 Tr])
70 ax = gca; ax.YAxis.Exponent = 3;

```

Simulering af sonar
 $A = 10$, $R = 43$ [m], $f_c = 450$ [Hz], $\sigma_r = 1$



Figuren viser, at den frekvensmodulerede puls stadig præcist detekterer afstanden til target.

Ikke overraskende går LP-filtrering værst ud over sinus-pulsen, som ikke længere kan give et præcist estimat på targets afstand. Den giver i stedet en “falsk-positiv” for denne samlede støjvektor. Med peak-detektion på sinus-systemet, ville man måle $\tau = 0.12$ s, svarende til $R = 21$ m. Dvs. en målefejl på 22m eller cirka 51 pct. for lavt.

Hvis der er usikkerhed omkring eller variationer i knækfrekvensen i bæremediet (her luften), er et frekvensmoduleret signal mere støjimmunt end en ren sinuspuls er.

3. Konklusion

Denne case har illustreret, hvordan digital signalanalyse kan benyttes i forbindelse med et sonar-system, der skal måle afstande til et target. Det er vist, hvordan forskellige signalparametre har indflydelse på et sonar-systems mulighed for præcist at måle afstande og at kunne skelne mellem forskellige targets (afstandsopløsning). Der er desuden diskuteret fordele og ulemper ved forskellige signaltyper. En række simuleringer er gennemført, med konklusionerne at:

- Et frekvensmoduleret signal (fx chirp) udviser puls-kompression, når det korreleres med et ekko af “sig selv”, hvilket giver høj præcision i afstandsmåling.
- Selv ved en væsentlig mængde støj er krydskorrelation en stærk og robust metode til at finde ét

signal i et andet signal.

- Et frekvensmoduleret signal er mere støjimmunt end et rent sinussignal.
- Når der er usikkerhed omkring hvorledes en kanal (her luften) vil filtrere et pulset signal, giver et frekvensmoduleret signal den bedste / sikreste performance. Fokus er vel ofte på at få maksimal performance ud af et “sonar”-signal, så det vigtigt at kunne presse systemet uden at risikere væsentlige fejl.

Der blev desværre ikke tid til at implementere et sonar-system i hardware (eller rettere: det blev nedprioriteret).

Det var en rigtig interessant case, med materiale der har mange interessante anvendelser.

4. Kildehenvisning

- [1] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. 1999. URL: <http://www.dspguide.com/pdfbook.htm>.
- [2] Roy Edgar Hansen. *Introduction to Sonar, INF-GEO4310*. 2013. URL: https://www.uio.no/studier/emner/matnat/ifi/INF-GEO4310/h13/undervisningsmaterieell/sonar_presentation_2013.pdf.
- [3] Dimitris G. Manolakis og Vinay K. Ingle. *Applied Digital Signal Processing: Theory and Practice*. 1. udg. Cambridge University Press, 2011.
- [4] RadarTutorial.eu. *Intrapulse Modulation and Pulse Compression*. 20?? URL: <https://www.radartutorial.eu/08.transmitters/Intrapulse%20Modulation.en.html>.
- [5] Wikipedia.org. *Pulse compression by linear frequency modulation (or chirping)*. 2020. URL: [https://en.wikipedia.org/wiki/Pulse_compression#Pulse_compression_by_linear_frequency_modulation_\(or_chirping\)](https://en.wikipedia.org/wiki/Pulse_compression#Pulse_compression_by_linear_frequency_modulation_(or_chirping)).

5. Funktioner

Der er til projektet benyttet følgende hjælpefunktioner.

5.1 setlatexstuff

```
1 function [] = setlatexstuff(intpr)
2 % Sæt indstillinger til LaTeX layout på figurer: 'Latex' eller 'none'
3 % Janus Bo Andersen, 2019
4     set(groot, 'defaultAxesTickLabelInterpreter',intpr);
5     set(groot, 'defaultLegendInterpreter',intpr);
6     set(groot, 'defaultTextInterpreter',intpr);
7     set(groot, 'defaultGraphplotInterpreter',intpr);
8
9 end
```