# Project 2 - Estimation Theory

INF4480

Petter Andre Kristiansen

February 11, 2018

University of Oslo, Department of Informatics

# Table of contents

# Statistical properties

# Exercise 1A Estimated PDF

$$\hat{m}_{\text{real}} = 0.2276, \;\; \hat{m}_{\text{imag}} = 0.0718$$
$$\hat{\sigma}^2_{\text{real}} = 27.1997, \;\; \hat{\sigma}^2_{\text{imag}} = 25.7465$$

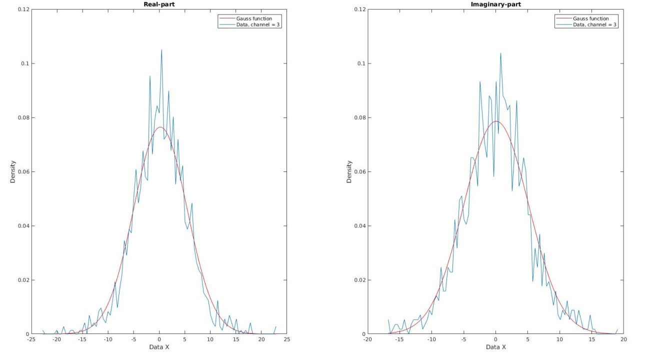



**Figure 1:** PDF of rawdata on channel 3

# Exercise 1B 1

**Are the 32 channels correlated?**

By looking at the matlab function *corrcoef*, we are able to check correlation on the whole data set. This returns a 32x32 complex matrix. Doing an *imagesc* of this, we can see its correlated with it self along the diagonal. Which is what we are looking for.

```
corr = corrcoef(data(1:1600,:));
figure;
subplot(121);
imagesc(real(corr));
title('Real');
subplot(122);
imagesc(imag(corr));
title('Imag');
```

# Exercise 1B 2

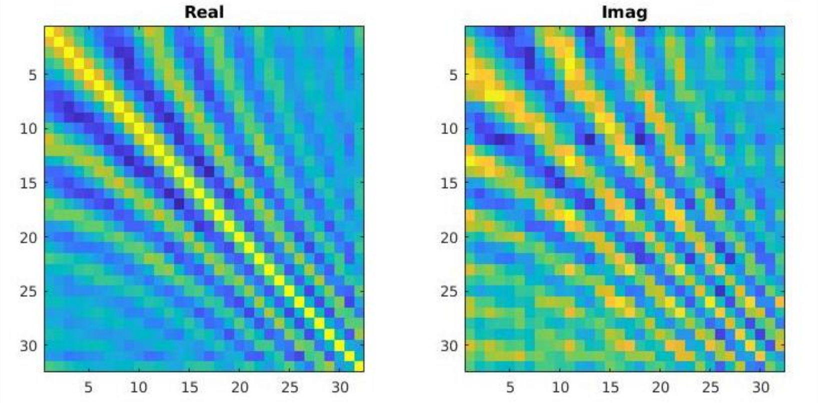



**Figure 2:** Cross-correlation over all channels

# Maximum likelihood estimation of time delay

# Exercise 2 A

**Derive the Maximum Likelihood Estimator**

We are looking at *EE522, Notes #14 Ch7C* from Binghamton University. Based on the signal model, we have:

$$x[n] = s[n - n_0] + w[n] \tag{1}$$

$s[n - n_0]$ becomes $s[s; n_0]$. Where $n_0$ is what we need to estimate.

# Exercise 2A MLE 1

$$x[n] = \begin{cases} w[n], & 0 \leq n \leq n_0 - 1 \\ s[n - n_0] + w[n], & n_0 \leq n \leq n_0 + M - 1 \\ w[n], & n_0 + M \leq n \leq N - 1 \end{cases} \tag{2}$$

Eq.(2) becomes our Maximum Likelihood Estimator as a product of each of the intervals:

$$p(x; n_0) = \prod_{n=0}^{n_0-1} Ae^{-\frac{x^2[n]}{2\sigma_x^2}} \cdot \prod_{n=n_0}^{n_0+M-1} Ae^{-\frac{(x[n]-s[n-n_0])^2}{2\sigma_x^2}} \cdot \prod_{n=n_0+M}^{N-1} Ae^{-\frac{x^2[n]}{2\sigma_x^2}} \tag{3}$$

Where $A = \frac{1}{\sqrt{2\pi\sigma_x^2}}$, $M =$ Pulse length and $N =$ Data samples.

# Exercise 2A MLE 2

Moving the constants out and grouping $x^2[n]$ terms:

$$p(x; n_0) = A^N \, \mathrm{e}^{-\frac{\sum_{n=0}^{N-1} x^2[n]}{2\sigma_x^2}} \cdot \mathrm{e}^{-\frac{1}{2\sigma_x^2} \overbrace{\sum_{n=n_0}^{n_0+M-1} \left(-2x[n]s[n-n_0] + s^2[n-n_0]\right)}^{\text{Expression of interest}}} \tag{4}$$

To be able to maximize the MLE, we need to maximize the expression of interest:

$$\sum_{n=n_0}^{n_0+M-1} \left(-2x[n]s[n-n_0] + s^2[n-n_0]\right) =$$

$$\sum_{n=n_0}^{n_0+M-1} -2x[n]s[n-n_0] + \sum_{n=n_0}^{n_0+M-1} s^2[n-n_0] \tag{5}$$

# Exercise 2A MLE 3

Now we can extend the summation limits because $s[n - n_0] = 0$ outside the current limits and pull the constant out. Eq.(5) can be rewritten as:

$$-2 \sum_{n=0}^{N-1} x[n]s[n - n_0] + \underbrace{\sum_{n=0}^{N-1} s^2[n - n_0]}_{\text{Energy} = \text{Constant}} \tag{6}$$

The second expression in eq.(6) is not dependant on $n_0$ when we extend the summation limits and we are left with maximizing:

$$\sum_{n=0}^{N-1} x[n]s[n - n_0] \tag{7}$$

# Exercise 2A MLE 4

The equation we are left with in eq.(7), is recognized as a cross-correlation.

**Cross-correlation** $C_{xs}$

To be able to maximize the MLE, we need to find the index where the cross-correlation is max.

$$C_{xs}[n_0] = \sum_{n=0}^{N-1} x[n]s[n-n_0] \tag{8}$$

$$\hat{n}_0 = \underset{n_0}{\operatorname{argmax}}\{C_{xs}[n_0]\} \tag{9}$$

# Exercise 2B

## Implement the ML Estimator for time delay

Using the equation (1) from project description as the transmitter signal:

$$s_{Tx}(t) = \begin{cases} e^{j2\pi\alpha t^2/2}, & -T_p/2 \leq t \leq T_p/2 \\ 0, & |t| > T_p/2 \end{cases} \tag{10}$$

The received signal $s_{Rx}[n]$ is given by the raw data.

# Exercise 2B Matlab-code 1

```
% 1600 first samples on channel 5
N = 1600; s_Rx = data(1:N,5).';
% Pulse length M
M = T_p*fs;
% Pulse time vector
t_chirp = linspace(-T_p/2, T_p/2, M);
% Chirprate
chirprate = B/T_p;
% Transmitted signal (chirp):
s_Tx = exp(1i*2*pi*(chirprate/2)*t_chirp.^2);
s_Tx = [s_Tx zeros(1, N-M)];
% Cross-correlation ( From Maximum Likelihood Estimator )
[Cn,lag] = xcorr(s_Rx, s_Tx);
```

# Exercise 2B Matlab-code 2

We are to look only at the postive side of the time delay response. An extra thing to notice, is that we get a $2 * N$ length correlation vector which we need to adjust to get the wanted result.

```
Cn = Cn(N:end);
lag = lag(N:end)/fs + t_0;
% Max value and index, peak detection
[val, ind] = max(abs(Cn));
% Fish located at seconds
fish = ind/fs + t_0;
% Plot of cross-correlation
plot(lag*1e3,abs(Cn));
```

Transforming lag into milliseconds, we need to divide the vector with the sampling frequency *fs* and multiply with 1000.

# Exercise 2B Matlab plot

We can see from figure 3, that we have a large spike at around 16 milliseconds, which represent an object like a fish.

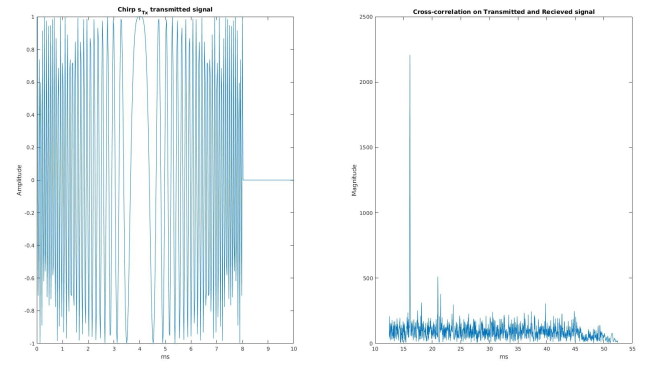



**Figure 3:** Chirp $s_{Tx}$ and Cross-correlation on channel 5

# Exercise 2C

The differences are the spikes in the compressed signals which represent the similarities in the received and transmitted signal

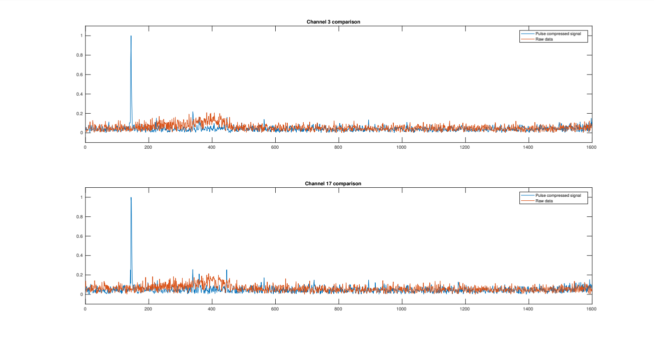
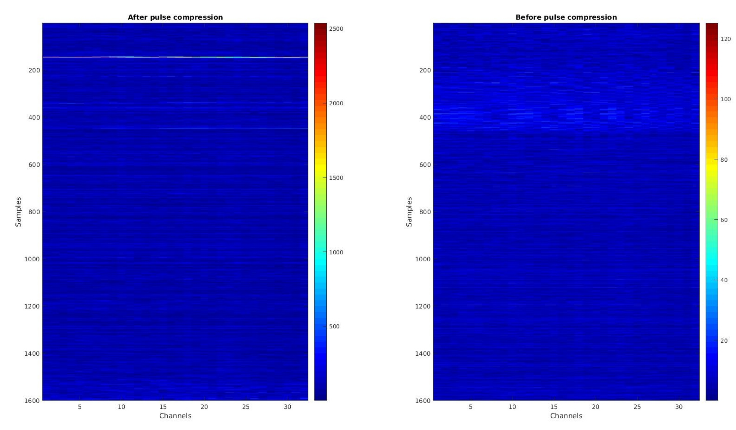



**Figure 4:** Puls compressed signal compared with raw data on channel 3 and 17

# Exercise 2B Cross-correlation $C_{xs}$ on all channels

The main difference on these two images is that we now can see points in the received signal which matches the transmitted signal in the pulse compressed signal. This brings out the points in the water column that reflects back earlier than expected, as for example from a fish.



# Exercise CB Matlab-code

Using a simple for-loop, we can iterate through all channels an create an image of the pulse compression.

```
s_Tx = exp(1i*2*pi*(chirprate/2)*t_chirp.^2);
s_Tx = [s_Tx zeros(1,N_t-M)];
compr = zeros(N_t,N_h);
for i = 1:N_h
    s_Rx = data(:,i).';
    [corr, lag] = xcorr(s_Rx, s_Tx);
    step = corr(1,N_t:end);
    compr(:,i) = step.';
end
imagesc(abs(compr))
colormap jet
colorbar
```

# Exercise 2D Estimated PDF

The PDF for the pulse compressed signal is not normalized, which results in the x-axis being larger. We see that the variances are slightly lower and which also tells us that we have a *precise* estimation.

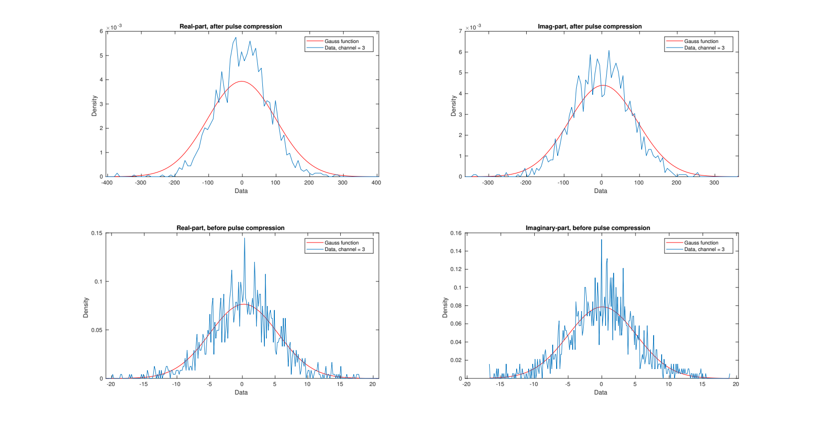



**Figure 6:** Compared PDFs of pulse compressed signal and 1A

# Cramér-Rao lower bound

# Exercise 3A

Cramér-Rao lower bound is simplified down to:

$$Var\{\hat{\tau}_0\} \geq \frac{1}{SNR \cdot \beta_{rms}^2}(sec^2) \tag{11}$$

$$SNR = \frac{max(C_{xs}[n_0])}{E\{C_{xs}[n_0]\}},\ \beta_{rms}^2 = (2\pi f_0)^2$$

The matlab-code equivalent to this is:

```
CRLB = @(SNR, f_0) (1/(SNR*(2*pi)^2 * f_0^2));
SNR = max(Cn)/mean(Cn);
CRLB_c = CRLB(SNR, 100000);
limit_c = sqrt(abs(CRLB_c));
```

# Exercise 3A CRLB on 2 different channels

The maximum limit of variation from the true value, is varying from the different channels. They are varying because the time it takes to get the reflected echo is consistent with how much or what it hits along its way. This can confirm that our measurements is correct because if the time delay were equal over all channels, we would suspect errors in the measurement data.

| Channels | Limit |
|---|---|
| Channel 6 | 64.48 nanoseconds |
| Channel 32 | 46.23 nanoseconds |

**Table 1:** Comparing CLRB on channel 6 and 32

# Exercise 3B

Time delay estimation for most dominant peak:

```
[peaks, channels] = max(abs(compr(1:N,:)));
[~, channel] = max(peaks);
max_index = channels(1,channel);
td_max = (max_index/fs + t_0)*1e3;
```

Resulting in a time delay estimate of

$$16.125 \ milliseconds$$

# Exercise 3B Time delay estimation on all channels

The results are not the same over all channels. We can see from last page that channel 12 was dominant, and that the object is most likely in the front section of the receivers.

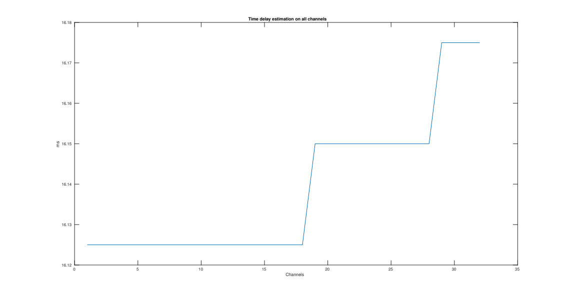



**Figure 7:** Time delay estimation on all channels

```
td_all = (channels./fs + t_0)*1e3;
plot(1:N_h, td_all);
```

# Least Squares Estimation

# Exercise 4A

An attempt to use the LS Estimator, I got an result with current parameters values:

$$\hat{\sigma}_\tau^2 = 0.3 nanoseconds,\ \hat{\tau}_0 = 0.028 milliseconds,\ \hat{l}_0 = 2076.7 Samples$$

This gave an estimate:

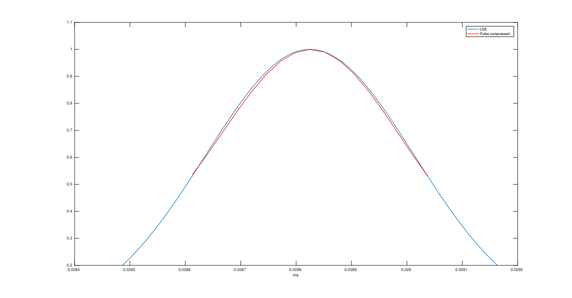



**Figure 8:** Least Squares Estimation of channel 8

# Exercise 4A Matlab-code

```
upr = 8;
compr = abs(compr(1:N,:));
rscompr = resample(compr,upr,1);
p8 = rscompr(:,8);
[max_p8, ind_p8] = max(p8);
p8_cut = p8(p8 > max_p8/2);
ax = size(p8_cut,1)/2;
t_s = linspace((ind_p8-ax)/fs,(ind_p8+ax)/fs,ax*2);
t_ax = linspace(t_0, N/fs +t_0, upr*N);
N_s = size(t_s,2);
H = [ones(N_s, 1) t_s.' (t_s.^2).'];
lse_theta = H\log(p8_cut);
th0 = lse_theta(1); th1 = lse_theta(2); th2 = lse_theta(3);
var_T = -(1/(2*th2));
tau = -(th1/(2*th2));
I0 = exp(th0 - ((th1.^2)/(4*th2)));
est = I0.*exp((-(t_ax-tau).^2)./(2*var_T));
```

# Exercise 4B

Due to problems on 4A, this exercise was hard to finish.

Questions?