INF4480 Project II: Estimation

Roy Edgar Hansen

January 24, 2021

Distribution date	January 29, 2021
Deadline	February 09, 2021, 10:15
Deliver presentation in	`Devilry`

Introduction

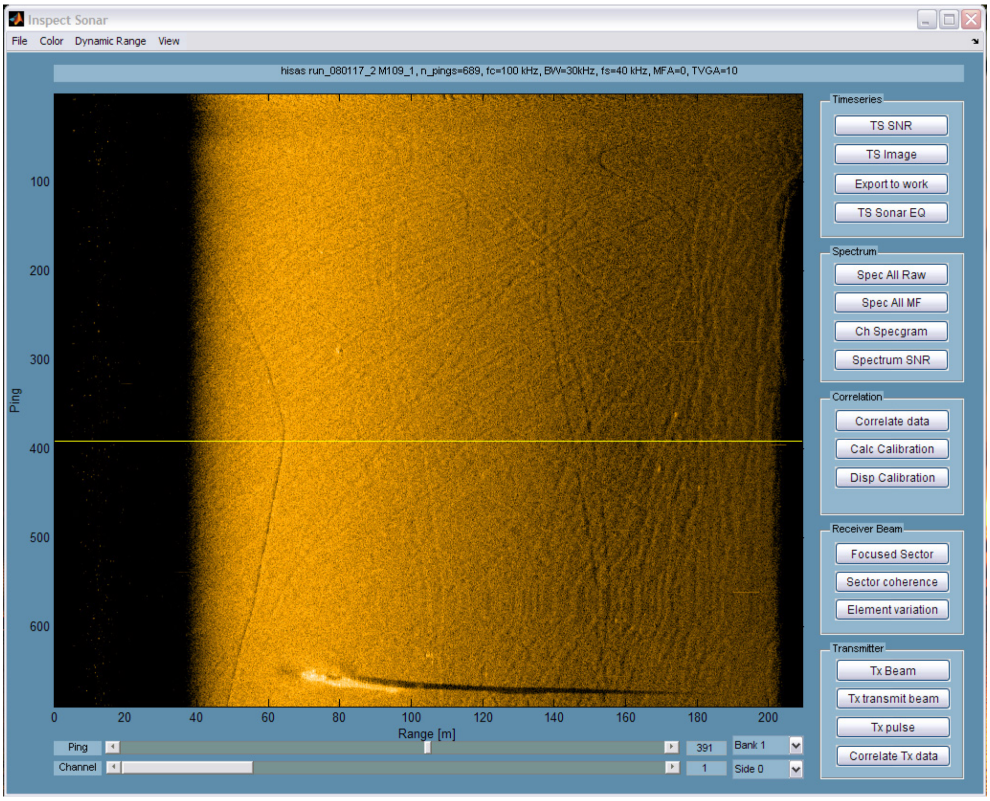

Figure 1: Sidescan sonar image

The exercises in this project will be related to sonar data collected by the HUGIN autonomous underwater vehicle. Figure 1 shows the recorded sonar data stacked in one line per ping and shown in a two-dimensional image as function of range (x-axis) and ping number (y-axis). This is called a sidescan sonar image.

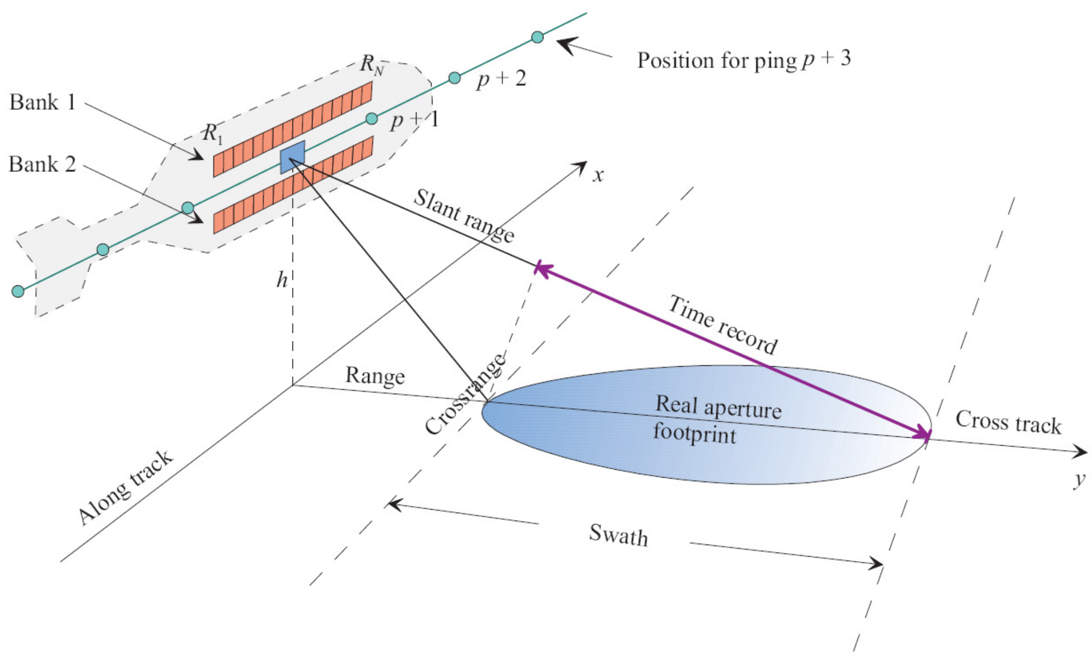

Figure 2: Sonar geometry

The sonar geometry is shown in Figure 2. We will consider a single ping (or pulse) with recorded timeseries from one horisontal receiver array with $N_h = 32$ hydrophones (receivers).

Our selected object of interest is a fish swimming in the water column at approximately 12 m range. Figure 3 shows a sonar image from a single pulse zoomed in on the relevant range interval. The sonar image is constructed by beamforming which is a key topic in the course IN5450 *Array Signal Processing*.

The data file `sonardata2.mat` at the course web page, is a matlab mat-file that contains the recorded timeseries from a single ping of sonar data as described above. The variable `data` contains a complex matrix of size $[N_t, N_h]$, where $N_h = 32$ is the number of hydrophones, and N_t is the number of time samples. The sampling frequency is stored in variable `fs` (in Hertz) and the start time for the recordings is stored in the variable `t_0` (in seconds).

The transmitted signal is a Linear Frequency Modulated (LFM) pulse of pulse length T_p, with signal bandwidth of B, as follows

$$s_{Tx}(t) = \begin{cases} \exp(j2\pi\alpha t^2/2) & -T_p/2 \le t \le T_p/2 \\ 0 & |t| > T_p/2 \end{cases} \tag{1}$$

where α is the chirp rate related to the signal bandwidth as

$$\alpha = B/T_p \tag{2}$$

The programmed signal bandwidth was $B = 30$ kHz, and the center frequency of the signal was $f_c = 100$ kHz. Note that the received timeseries are basebanded at reception (the carrier frequency has been removed). We have therefore taken out the carrier (center frequency) from the signal model.

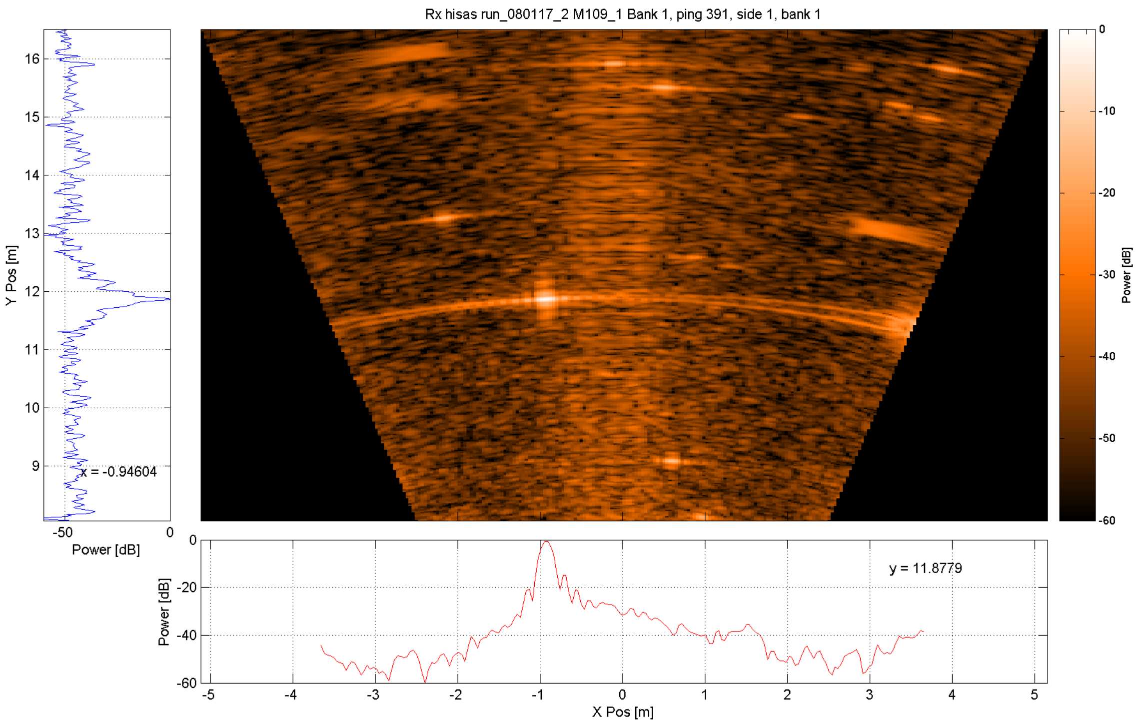

Figure 3: Sectorscan sonar image zoomed in on a small object

1 Maximum likelihood estimation of time delay

Exercise 1A
Assume the signal model presented in the *time delay* example in Lecture 5 (Estimation II). Derive the Maximum Likelihood Estimator for time delay estimation. Follow the `Notes #14 Ch7C` in the *EE522* lecture notes from Binghamton University.

Exercise 1B
Implement the ML Estimator for time delay. Hint: Use the transmit signal (1), and the matlab function xcorr. Only use the result for positive delay. Matlab code has to be written in the presentation.

Exercise 1C
Apply the the cross correlation from the ML estimator in the previous exercise on all the received data (all channels, all time samples). The output sequences are the so-called pulse compressed version of the received signals. This process is also known as matched filtering. Hint: Use the complex version of the transmit signal, and apply on the complex receiver signals. The output pulse comresssed signals should also be complex. Normalisation is not important.

Choose the first 1600 samples in the pulse compressed time series (equivalent to the signals reflected from the water column). Plot the magnitude (or absolute value) of the complex sequence before and after pulse compression for a selected receiver channel. Explain the differences.

Exercise 1D
Estimate the time delay for the most dominating peak in the selected interval (the first 1600 samples), by using the ML estimator from exercise 1B. Do the peak detection on the magnitude (or absolute value) of the complex sequence.

Plot the estimated time delay for all the channels. Upsample the complex sequences by a factor 4 or 8 before peak detection. Plot the results in the same figure.

Are the results the same as function of receiver channel? Explain.

Are the results the same before and after upsampling? Explain.

2 Cramér-Rao lower bound

Exercise 2A
Follow the example from Lecture 5, and calculate the Cramér-Rao lower bound for time delay estimation. Base the calculations on the pulse compressed signals on the first 1600 samples. Choose the strongest reflector as signal, and estimate the noise level based on the rest of the samples. Choose yourself the approach to calculate the lower bound. Note that the pulse compressed version of the signal has compressed the pulse length from T_p to (ideally) $1/B$.

Calculate the CRLB for all the channels.

Are the results the same? Explain.

3 Least Squares Estimation

Exercise 3A
Implement the Linear Least Squares Estimator for time delay estimation. Follow the Time delay example in the *least squares estimation* in Lecture 5. Use the magnitude squared (to reflect intensity) of the pulse compressed signals of the first 1600 samples. Upsample by a factor of 4 or 8. Hint: see the function `resample` in matlab. Apply the method given in the lecture to produce the linear least squares fine (sub-sample) time delay estimates. Plot the result for one pulse.

Exercise 3B
Plot the estimated time delay for all the channels using the LSE in 3A.

Are the results the same for each channel? Explain.

Compare with the estimated time delays in Exercise 1D. Discuss the differences.