Statistical Signal Processing

Question 1

Tau was found by measuring the cross-correlation between the transmitted signal, u(t), and the received signal, y(t).

$$y(t) = au(t - 2\tau) + n(t)$$

As the probability distributions of the transmitted and received signals were unknown, the cross-correlation was estimated. This was done using the following equation, where *d* represents the unknown lag:

$$r_{uy}(d) = \sum_{t=1}^{1000} y(t)u(t-d) = \sum_{t=1}^{1000} [au(t-2\tau)u(t-d) + n(t)u(t-d)]$$

As the noise and the transmitted signal are uncorrelated, the second term will have little contribution to the cross-correlation. Therefore, the cross-correlation would be maximized when the unknown lag was equal to the delay, 2τ. The cross-correlation time series for the data provided is shown by figure 1. The maximum value of the cross-correlation was 886 and that occurred when the lag had the value 2. The lag, $d = 2 = 2\tau$, corresponds to the time taken for the signal to reach the object and be reflected; therefore, $\tau = 1$ must be the time taken for the signal to reach the object.

The *Matlab* code used to implement this method is included in the appendix. The function *xcorr* was used to compute the cross-correlation and the *max* function was used to find the maximum value and its corresponding index of the cross-correlation vector. The index of the maximum cross-correlation values maps to the index of the lag which produced it.

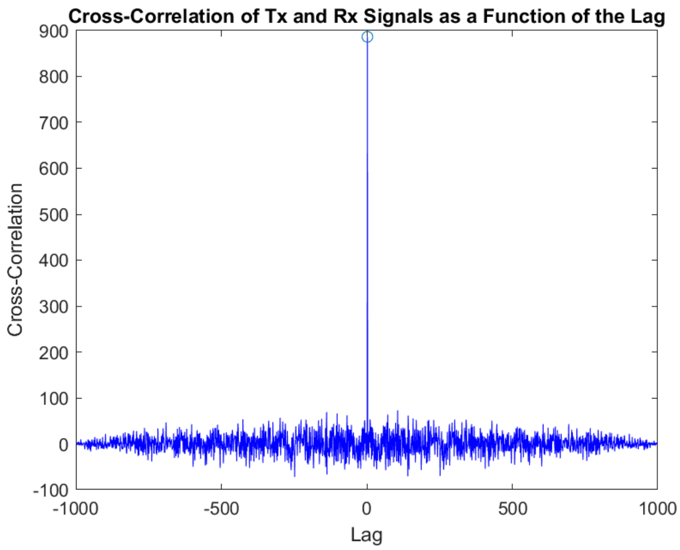

FIGURE 1: CROSS-CORRELATION AS A FUNCTION OF THE UNKNOWN LAG

Question 2

The received signal, $x[n]$, was modelled using the following equation:

$$x[n] = s[n - n_0] + w[n], \quad where\ s[n - n_0] = \begin{cases} 0, & 0 \le n \le n_0 - 1 \\ a, & n_0 \le n \le n_0 + M - 1 \\ 0, & n_0 + M \le n \le N - 1 \end{cases}$$

where *a* represents the height of the transmitted signal's pulse and w[n] is a normally distributed random variable with zero-mean and variance, σ^2. Since w[n] represents AWGN, it was assumed each sample was a realisation of an independent but identically distributed random variable. Given the pdf of w[n], the pdf of x[n] is also normally distributed with mean s[n-n_0] and variance, σ^2. Since w[n] are independent, so are x[n]; however, they are not identically distributed because the mean of each x[n] is dependent on the value of $s[n - n_0]$. The time index, *n*, and the parameters *N* and *M* are known which leads to the joint pdf conditioned upon n_0 being as follows:

$$p_{x|n_0}(x; n_0) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\{\frac{-1}{2\sigma^2} \sum_{n=0}^{N-1} (x[n] - s[n - n_0])^2\}$$

This is known as the likelihood function as it represents the likelihood of the observed data given n_0. Writing this in vector form leads to:

$$p_{x|n_0}(x; n_0) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\{\frac{-1}{2\sigma^2} (x - s[n_0])^T (x - s[n_0])\}$$

where $x^T = [x[0], x[1], \dots, x[N-1]]$ and $s[n_0]^T = [s[-n_0], s[1 - n_0], \dots, s[N - 1 - n_0]]$.

The maximum likelihood estimation finds the value of n_0 which maximises the probability of the observed data. This is formulated as follows:

$$\hat{n}_0 = \arg\max_{n_0} p_{x|n_0}(x; n_0)$$

Each value of *s* is given by a constant, hence differentiating the pdf with respect to n_0 will be fruitless. However, *s* is the only variable in $p_{x|n_0}(x; n_0)$ which is dependent on n_0. Therefore, the constant can be ignored and only the term within the exponential considered. Since the exponential is negative, the maximum likelihood estimation can be equally represented by minimising the sum of squares:

$$\hat{n}_0 = \arg\min_{n_0}\{(x - s[n_0])^T (x - s[n_0])\}$$

This can be thought of as finding which delay to the input signal causes it to most closely approximates the received signal. As mentioned in the problem description, for all possible values of n_0 the delayed signal $s[n - n_0]$ is contained within the observational interval; as a result, n_0 must be in the range $0 \le n_0 \le N - M$.

The transmitted and received signals described in part 2 are illustrated by figure 2 where the delay $n_0 = 50$ was used for the deterministic component of the received signal. Outside of experimentation, n_0 would be unknown. To find n_0 using the maximum likelihood estimator, the sum of squares was used as the objective function for all possible n_0 and the n_0 corresponding to the minimum value is used as the estimator. For the signals shown in figure 2, the related objective function is shown by figure 3 where the minimum had the value 14.5 and corresponded to a lag of 52. This has an accuracy of 96% where the deviation was caused due to the AWGN.

The code used to implement the maximum likelihood estimator is included in the appendix. It was implemented using the *circshift* function to cycle through the time shifted waveforms and for each waveform the sum of squares was computed and stored. The *min* function was used to identify the optimal lag and sum of squares value.

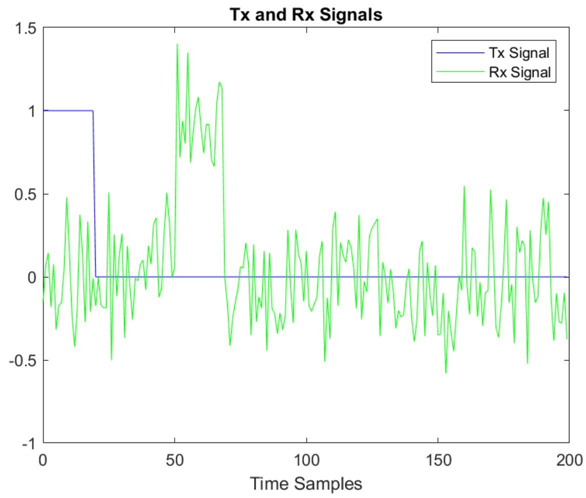

FIGURE 2: TRANSMITTED AND RECEIVED SIGNALS

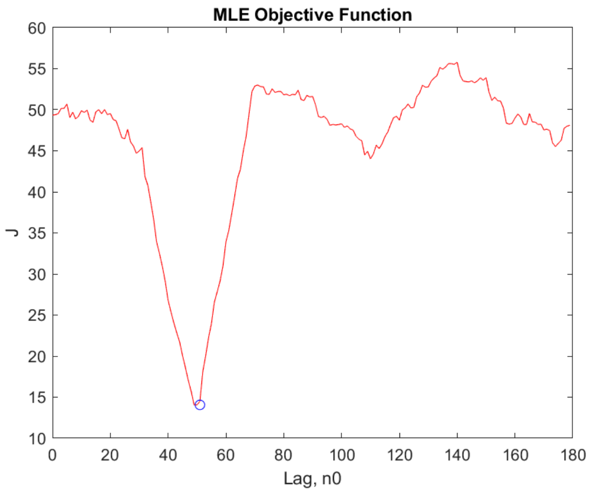

FIGURE 3: SUM OF SQUARES AS A FUNCTION OF THE LAG

```
clear all; close all; clc;
load('delay.mat');

[r, lags] = xcorr(y,u); %returns cross correlation and their corresponding lags
[max, max_index] = max(r); %finding maximum argument of cross-correlation array
max_lag = lags(max_index); %lag which maximised cross-correlation

figure(1);
plot(lags,r,'b');
hold on
scatter(max_lag, max);
title('Cross-Correlation of Tx and Rx Signals as a Function of the Lag');
xlabel('Lag');
ylabel('Cross-Correlation');
```

```
clear all; close all; clc;

% discrete time
N =200;
n = transpose(0:N-1);

% Tx signal s[n]
M = 20;
height = 1;
s = zeros(200,1);
s(1:M) = height;

% Rx Signal with additive noise normally distributed, 0 mean and 0.25 variance
n0 = 50; % delay
x = zeros(200,1);
x(n0+2:n0+M+1) = height;
x = x + 0.25*randn(200,1);

% Plotting data
figure(1);
plot(n,s,'b');
hold on
plot(n, x,'g');
title('Tx and Rx Signals');
xlabel('Time Samples');
legend('Tx Signal','Rx Signal');

% MLE
s_data = zeros(200,1);
J_vect = {};

for n0_est = 0:N-M-1
    s_data = circshift(s,n0_est);
    J = transpose(x-s_data)*(x-s_data);
    J_vect = [J_vect, J];
end

% find index of J_vect which corresponds to its minimum value
J_vect = cell2mat(J_vect);
[min, min_index] = min(J_vect)

% plotting MLE criteria
n0_est = 0:N-M-1;
figure(2);
plot(n0_est,J_vect,'r');
hold on
scatter(min_index,min,'bl');
title('MLE Objective Function');
xlabel('Lag, n0');
ylabel('J');
```