Radar simulation

David Collins

October 7 2013

Contents

1 Overview

In this laboratory we tried to simulate a radar application using cross correlation of the transmitted and reflected signals. Radar uses waves of electromagnetic radiation as signals (radio waves more specifically), but similar principles apply to sonar (which uses sound rather than electromagnetic radiation).

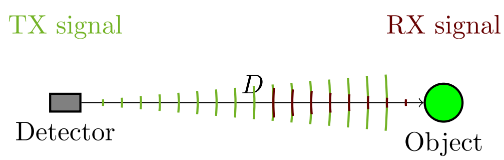

Figure 1: Using radar to determine distance to an object

2 Theory

Before proceeding to solve the given problem, it is worth trying to understand the theory behind it briefly.

2.1 Simplified radar application

We have an object which is a distance D away from our signal source. We don't know what D is, but want to find out.

$$v = \frac{D}{t}$$

$$D = vt$$

2.2 Cross correlation

Naturally we expect there to be a large correlation between a transmitted signal and its reflection.[1] The reflected signal will inevitably be 'offset' relative to the transmitted signal, however, due to the delays incurred in any physical medium. Thus, if we perform the usual correlation process on the two signals we will not necessarily find any meaningful information.

The correlation will be most evident if we 'shift' the received signal so that it 'lines up' with the transmitted one. In fact, the largest degree of correlation will be observed when the number of samples shifted is precisely equal to the number of samples that the received signal is offset relative to the transmitted signal. This phenomena is the basis behind the idea of 'cross-correlation'.

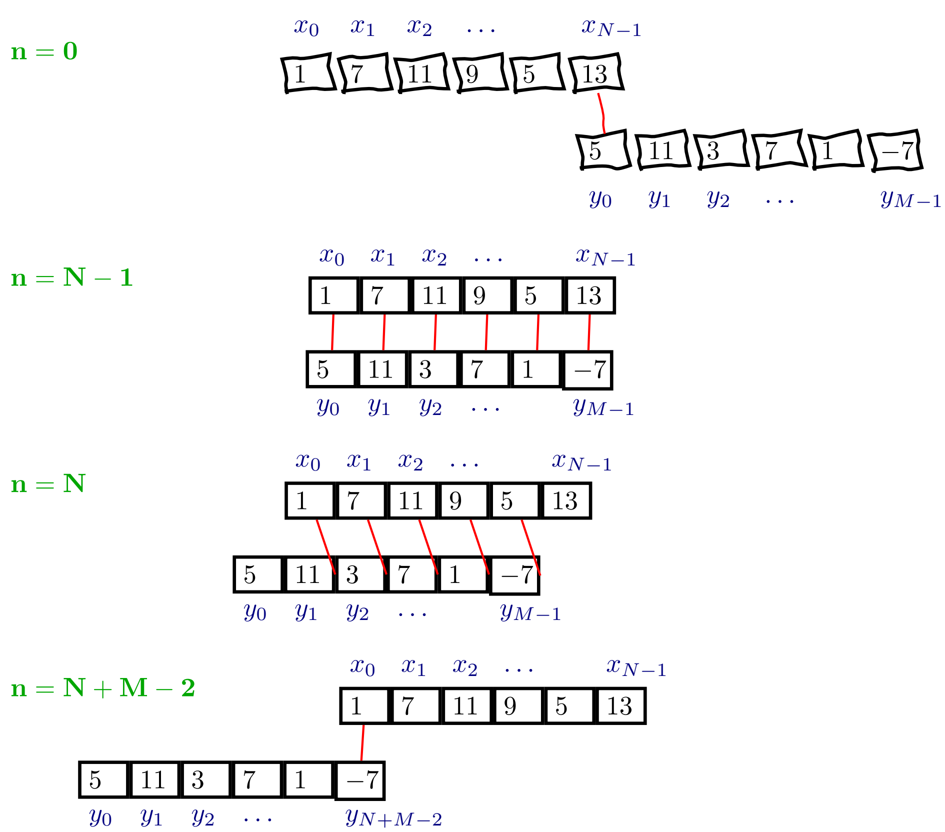

Figure 2: Calculating the cross-correlation for various values of n

Figure 2 shows snapshots of the cross-correlation process for different values of n. It can be established that the

[1] Correlation is an engineering term for 'similarity'.

cross-correlation of two sequences $x[n]$ and $y[n]$ is

$$r_{xy}[n] = \begin{cases} \sum_{k=0}^{n} x[N-1-n+k]y[k], & n \leq (N-1) \\ \sum_{k=0}^{N-1} x[k]y[k+n+1-N], & (N-1) < n < (N+M-1). \end{cases}$$

If there is zero delay between the transmitted and received signals (effectively impossible in a real-world application), the cross-correlation array will have maximum amplitude exactly in its middle element. This is perhaps best demonstrated by example (Figure 3).

Listing 1: Correlation of two similar signals with no offset between them

```
% Example of cross correlation when there is no delay
% between signals.
N = 2^5;

x = zeros(1,N);
x(7:11) = [5 1 13 -3 7];

y = 3*x;

cross_correlation = xcorr(x,y);

subplot(2,2,1);
stem(x);
xlabel('Sample number');
grid();
ylabel('Amplitude');
subplot(2,2,2);
stem(y);
xlabel('Sample number');
grid();
ylabel('Amplitude');
subplot(2,2,3:4);
stem(cross_correlation);
xlabel('Sample number');
text(length(cross_correlation)/2, max(cross_correlation)/2, 'No delay -> max occurs at center');
grid();
ylabel('Amplitude');

print -dpdf 'nodelay.pdf';

pause();
```

Next, lets introduce a delay (or 'offset') in the second signal.

```
y = shift(y,-offset);
```

The more delayed the signal is, the more the maximum amplitude in the cross-correlation array will be offset from the center. Can we be sure that this relationship is linear however? Let's try to find out. We can do this either analytically or empirically. We will go with the latter option here.

Listing 2 demonstrates one way of investigating this relationship.

Listing 2: Investigating the relationship between delay in received signal and nature of cross-correlated array

```
% Is there a linear relationship between the delay in the
% received signal and the offset of the peak amplitude in the
% cross-correlation of the two signals?
for i = 1:N/2
```

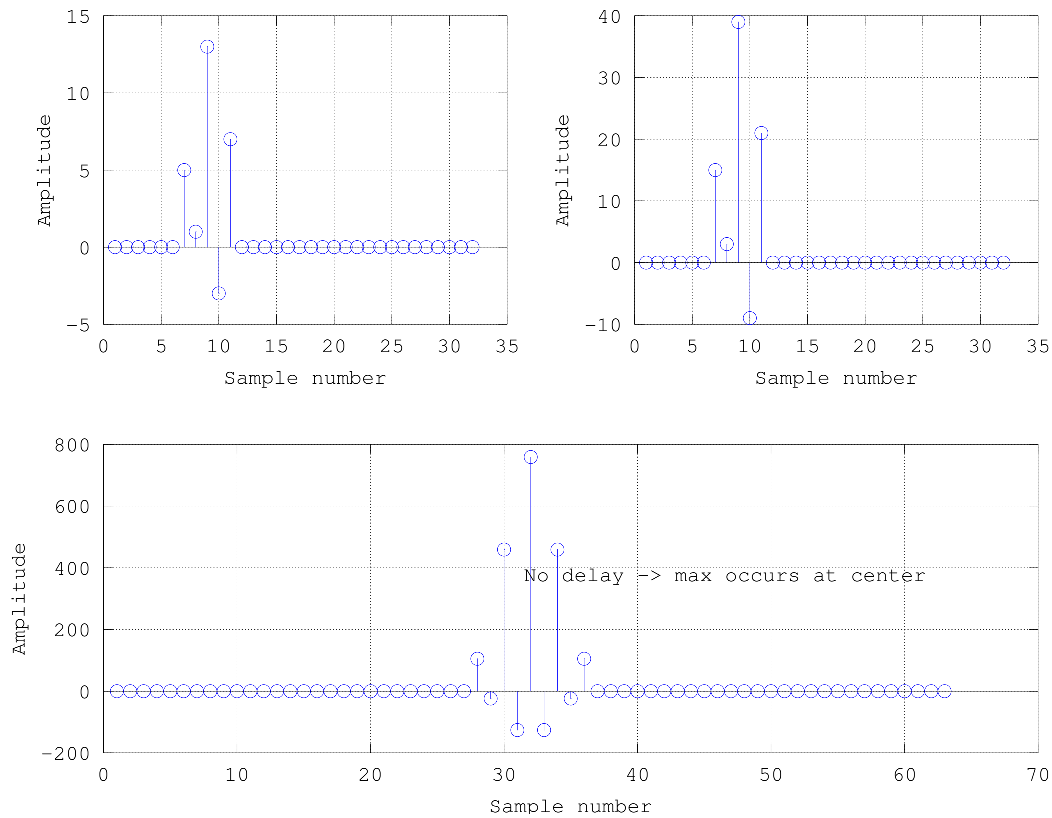

Figure 3: Cross-correlation of similar signals with no offset between corresponding samples

```
        delay_in_samples = i;
        tx_rand = randn(1,N);

        rx_rand = randn(1,N)*.2 + shift(tx_rand,delay_in_samples);

        cross_correlation = xcorr(rx_rand,tx_rand);

        [peak_value,peak_index] = max(cross_correlation);
        center_index = round(length(cross_correlation)/2);
        offset_from_center_in_samples = peak_index-center_index;

        rx_offset(i) = delay_in_samples;
        xcorr_offset_from_center(i) = abs(offset_from_center_in_samples);
end
```

The results are shown in Figure 5. As can be clearly seen, the relationship between the delay in the received signal and the offset of the peak amplitude in the cross-correlation of the transmitted and received signals is clearly linear.

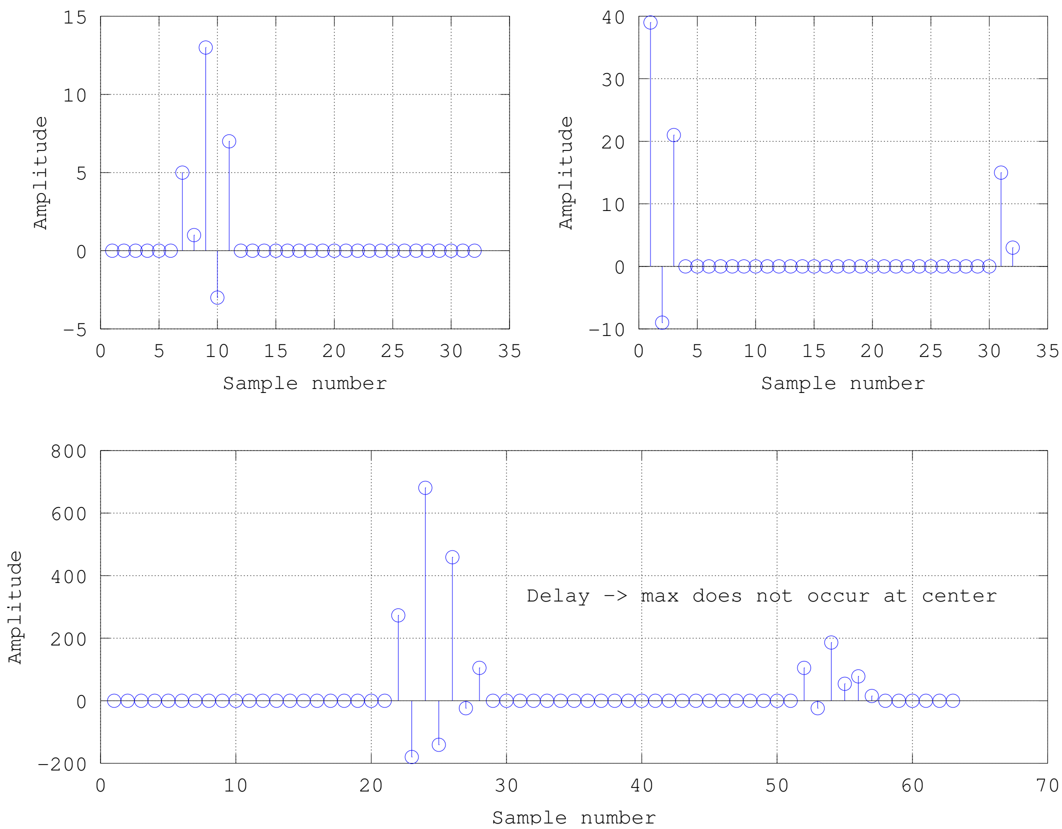

Figure 4: Cross-correlation of similar signals with an offset between corresponding samples. Offset = 11

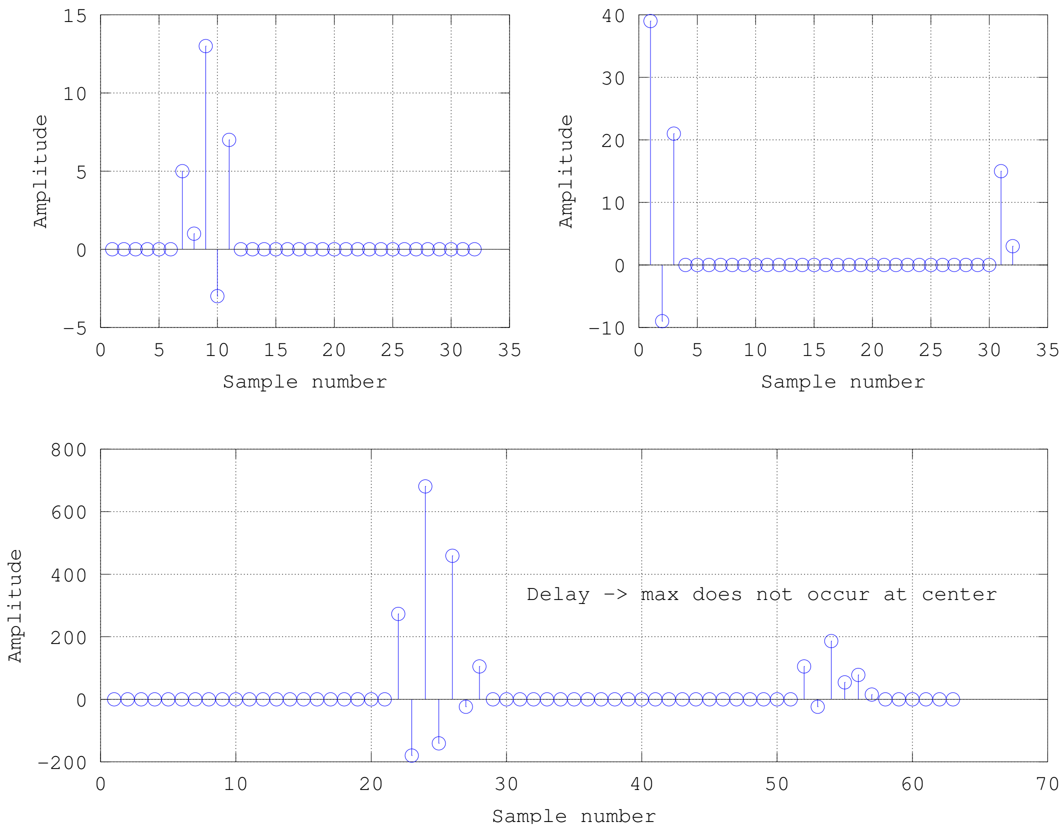

Figure 5: Demonstration of linear relationship between delay in received signal and position of peak amplitude in cross-correlated array

3 Implementation

The code for finding the delay between our transmitted and received signal is shown in Listing 3.

Listing 3: Finding the delay in the received signal

```
load('signals/tx_sinusoid');
load('signals/rx_sinusoid');
%load('signals/tx_rand');
%load('signals/rx_rand');

%rx = rx_rand;
%tx = tx_rand;
%rx = rx_unit;
%tx = tx_unit;
rx = rx_sinusoid;
tx = tx_sinusoid;

cross_correlation = xcorr(rx,tx);

[peak_value,peak_index] = max(cross_correlation);
center_index = round(length(cross_correlation)/2);
offset_from_center_in_samples = peak_index-center_index;

delay_in_samples = offset_from_center_in_samples

%printf("The delay (in samples) is %d", delay_in_samples);

subplot(2,2,1);
```

4 Results

4.1 Reflected signal delay

The results of the simulation — for different transmitted signals — are shown in Table 1. As can be seen, the calculated delay is almost the same in the case of both the sinusoidal signal and the random / noise signal. Although not conclusive, this increases our confidence that our results are plausible.

Signal	Calculated delay
Unit impulse	
Sinusoidal	2934
Random / noise	2933

Table 1: Results

4.2 Distance

To determine the actual distance to the object, we need to know the speed of wave propagation (that of light in this case), and the sampling rate. We will assume the latter to be $f_s = 1 \cdot 10^6 \mathrm{s}^{-1}$. Thus, the sampling period, T, is

$$T = \frac{1}{f_s} = 10^{-6}\mathrm{s}.$$

The time t, then, is

$$t = NT = 2.933 \cdot 10^3 \cdot 1 \cdot 10^{-6}\mathrm{s} = 2.933 \cdot 10^{-3}\mathrm{s}.$$

We calculate our distance, D, as follows.

$$c = \frac{D}{t} \therefore D = ct = (3 \cdot 10^8 \text{m/s})(2.933 \cdot 10^{-3} \text{s})$$
$$= 8.799 \cdot 10^5 \text{m}$$
$$\approx 880 \text{km}$$

4.3 Summary of results

Assuming a sampling rate of 1 MHz, our results are

$$\boxed{N = 2933}, \quad \boxed{D \approx 880\text{km}}.$$

5 Summary

In this lab we investigated the use of cross-correlation in a simplified radar simulation. We found that accurate results can be achieved using simple concepts.

5.1 Suitable signals

Based on our findings in Section 4, we can conlude the following:

- A noise signal is ideal for use as a transmitted signal in a radar application — as long as we keep a copy of the original. This is because a noise signal cross-correlated with itself yields a very strong peak at a single point.
- A sinusoidal signal also yields useful results — but not quite as accurate as in the case of a noise signal. More specifically, the precise location in the cross-correlation array at which the maximum amplitude occurs may have a certain amount of uncertainty associated with it. How serious a problem this is depends on the application.
- A unit impulse, as can be seen from Table 1, is not very useful at all.

A A. MATLAB code

The MATLAB code shown in Listings 1 and 3 was used primarily for analysis. The following code (Listing 4) was used to interact with the actual radar simulation.

Listing 4: Performing the actual radar simulation

```
% Remove unused variables from workspace
clear;

% Signal lengths
M = 2^10;
N = 2^8;

% window
%win = window(@hamming, N);

% Angular frequency of sinusoidal transmitted signal
tx_sinusoid_angular_frequency = 1/11;

tx_rand = randn(1,M);
% Other possible test signals
tx_unit = [1 zeros(1,N-1)]; % a unit impulse
n = 1:N;
```

```
tx_sinusoid = sin(2*pi*tx_sinusoid_angular_frequency*n/N);
% Your simulation variable / number = 1
simv = 1;
rx_rand = radar_sim(tx_rand, simv);
rx_unit = radar_sim(tx_unit, simv);
rx_sinusoid = radar_sim(tx_sinusoid, simv);

% Cross-correlation
xcorr_rand = xcorr(rx_rand,tx_rand);

% Plot results
subplot(2,2,1);
plot(tx_rand);
grid();
xlabel('Sample number');
title('TX');
subplot(2,2,2);
plot(rx_rand);
grid();
xlabel('Sample number');
title('RX');
subplot(2,2,3:4);
plot(xcorr_rand);
grid();
title('Cross-correlation of RX and TX');

pos = max(xcorr_rand);

save 'signals/rx_rand' rx_rand;
save 'signals/rx_unit' rx_unit;
save 'signals/rx_sinusoid' rx_sinusoid;
```