Prac 1: Octave

Roan J. Song[†]
EEE4084F Class of 2017
University of Cape Town
South Africa
[†]SNGROA001

***Abstract*—**

I. Introduction

Correlating two signals that are identical in shape but differing in phase/time introduces new questions: will the two signals still show perfect correlation? How will the correlation coefficient be affected by a shift in time, and how drastically? Does the sample rate of the signal have an effect?

II. Methodology

In this section you should describe the method of the experiment.

A. Hardware

The experiments conducted in this report were performed on a computer with the following specifications:

- CPU: i5-2500 (4 cores @ 3.3-3.7 GHz)
- RAM: 8 GB DDR3
- OS: Ubuntu 15.04

B. Introductory Experiments

To breed foster familiarity with Octave, a task was issued that involved generating a .wav file with a given method, and then creating one for comparison. The difference between the two functions being that the first one relies on a single call to rand() to generate multiple values, while the new method makes a call to rand() for each value in turn. A sample rate of 8000Hz was used, to decrease computation time.

The given implementation took 173ms to generate 1000s of 8000Hz white noise. The custom implementation took 64244ms. This new method is a handy 370 times slower than the original method. This speed-down is likely due to the overheads of calling the rand() function multiple times.

The code for the custom implementation is shown below:

```
function whiten = createwhiten(time)
  sample_rate = 8000;
  whiten = zeros(time*sample_rate,1);
  samples = time*sample_rate
  for i = 1:samples
    whiten(i) = rand();
  end

  whiten = whiten*2-1; %fit into .wav bounds
end
```

C. Custom Correlation Implementation

The source code for the bespoke correlation solution is shown below.

```
function r = mycorr(x,y)
        % readability >> succinctness
        sx  = sum(x);
        sy  = sum(y);
        sxy = sum(x.*y);    % elementwise multiplication
        sx2 = sum(x.^2);    % elementwise squares
        sy2 = sum(y.^2);
        n   = max(size(x)); % just in case we get some column/row vector mixups
        num = sxy - sx*sy/n;
        den = sqrt((sx2-(sx.^2)/n)*(sy2-(sy.^2)/n));
        r = -2; % default, in case correlation could not be performed
        if(den != 0)
                r = num/den;
        end
end
```

D. Comparing Signals Shifted in Time

Firstly, a signal must be generated, and then compared against a shifted version of itself. This can easily be done by generating a signal and then using a small 'window' of the signal compared to a window shifted slightly to either side, i.e. if the signal x is comprised of 100 samples, one signal could be x[1:50] and the shifted version 20 samples ahead, at x[21:70]. For all instances, I used a window size of 50 samples.

The signals generated were simple sine waves, at frequencies of 1rad/s, 2rad/s, and 10rad/s. All of the signals generated range from -pi to pi (shown in Figure 1). 100, 1000, and 10000 samples were used for each frequency. The choice of windows are [1:50] (the first 50 samples of the signal), [2:51] (a shift by 1), and [11:61] (a shift by 10).

These values were chosen to cover a wide range of signals, and to avoid jumping to conclusions.

Figure 2 shows a low frequency, low sample rate signal viewed through the different windows. Figure 3 shows a high frequency, low sample rate signal viewed through the different windows. Figure 4 shows a high frequency, high sample rate signal viewed through the different windows. Figure 5 shows the absolute value of the differences between shifted signals at different sample rates.

Judging from these figures the inference can be drawn that low sample rate signals will be more heavily affected by a shift in samples, as there is a larger gap of information between each sample.

The correlation between each signal was calculated using the native corr function as follows:

```
a = linspace(-pi,pi,100); %100 samples
x1  = sin(a*1); %frequency of 1rad/s
w1 = 1:50;
w2 = 2:51;
r = corr(x1(w1),x1(w2));
disp(r);
```

E. Experiment Procedure

Furthermore, include detail relating to the experiment itself: what did you do, in what order was this done, why was this done, etc. What are you trying to prove / disprove?

III. Results

A. Custom Correlation

It appears that the custom correlation implementation performs faster than the native Octave algorithm. The average speedup varies between ~16 on smaller vectors to just above 1.2 on very long (10^8) vectors.

Sample Size	mycorr (ms)	corr (ms)	Speed-up
2.4128	3.7909	6.3648e+00	10
2.1110	1.3089	1.6128e+01	100
2.1570	1.5306	1.4092e+01	1000
2.4791	2.6298	9.4270e+00	10000
3.7451	1.3590	2.7558e+00	100000
21.465	11.776	1.8228e+00	1000000
140.40	104.61	1.3422e+00	10000000
1344.2	1042.5	1.2895e+00	100000000

TABLE I
TABLE COMPARING TWO CORRELATION IMPLEMENTATIONS

B. Effects of Time Shifts on Correlation

100 samples	1 rad/s	2 rad/s	10 rad/s
shift by 1	0.989	0.991	0.802
shift by 10	0.494	0.294	0.998
1000 samples	1 rad/s	2 rad/s	10 rad/s
shift by 1	1.000	1.000	0.990
shift by 10	1.000	0.999	0.479
10000 samples	1 rad/s	2 rad/s	10 rad/s
shift by 1	1.000	1.000	1.000
shift by 10	1.000	1.000	1.000

IV. Conclusion

I further infer that a shift in seconds instead of samples would provide the same results for each frequency, as the values at each point would be the same given that the point was defined for each signal.

References

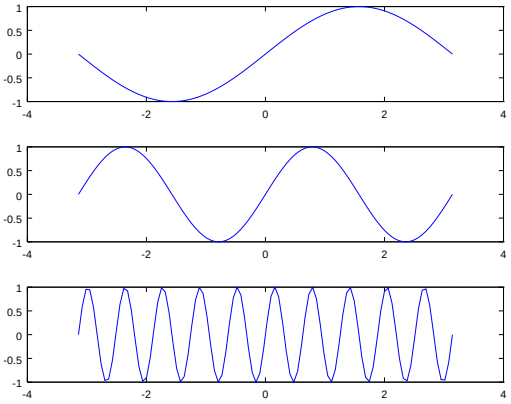

Fig. 1. Overview of signals

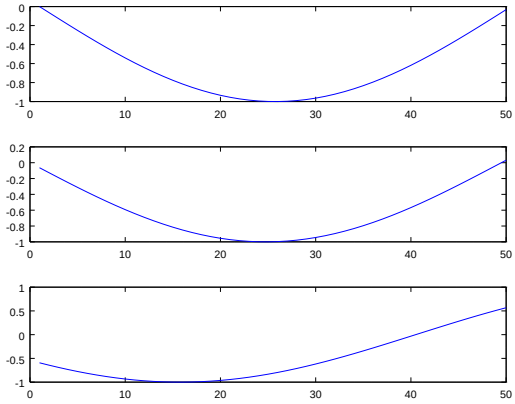

Fig. 2. Overview of signals

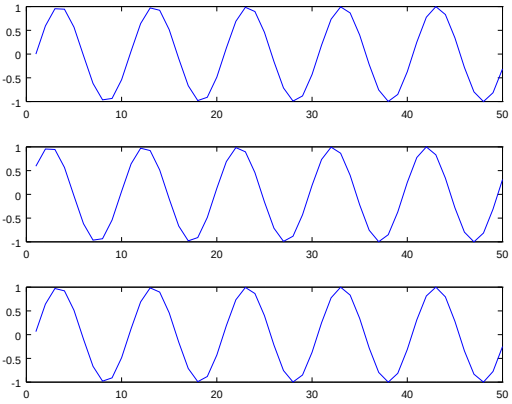

Fig. 3. Overview of signals

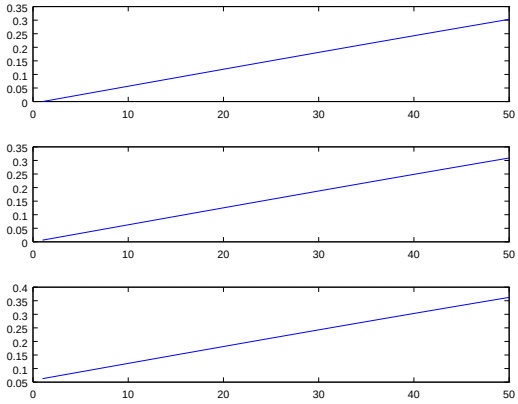

Fig. 4. Overview of signals

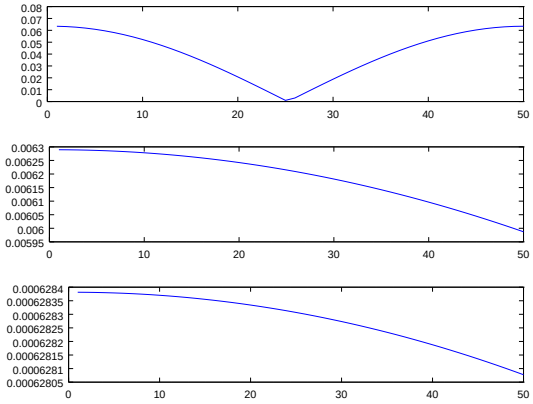

Fig. 5. Overview of signals