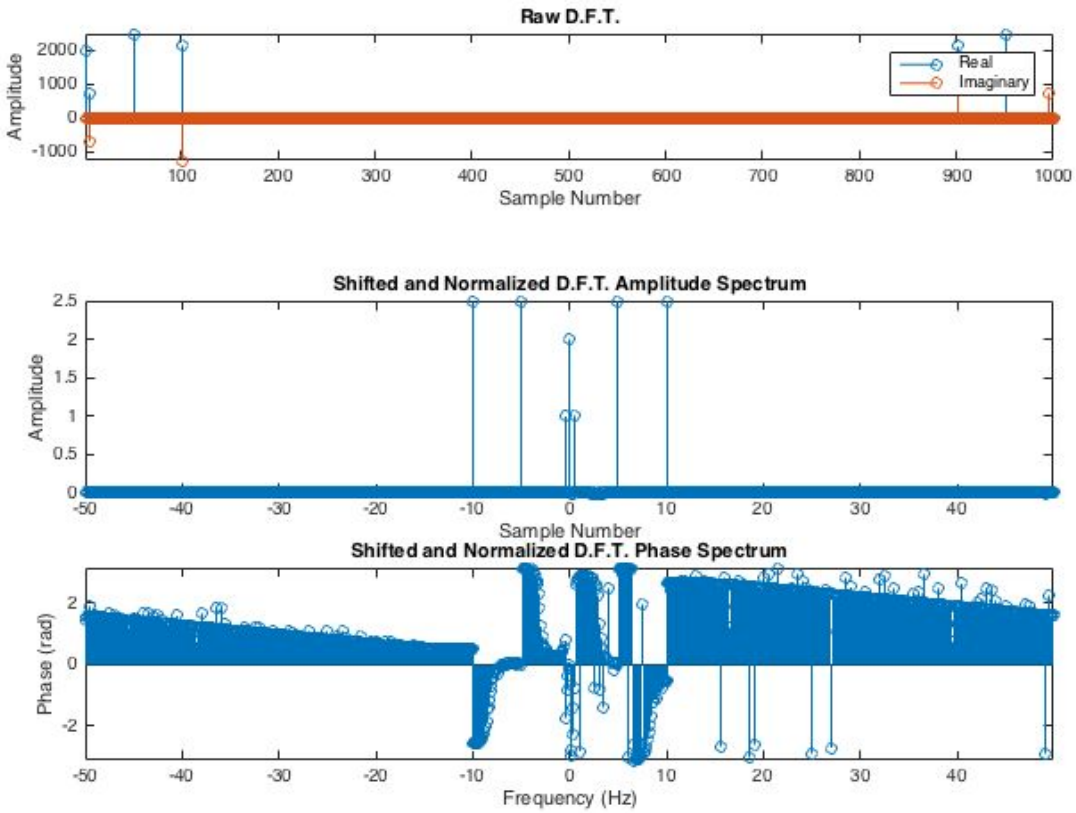
Garrett Sickles
GPGN 404
6 November 2015

Lab 5: DFT and IDFT

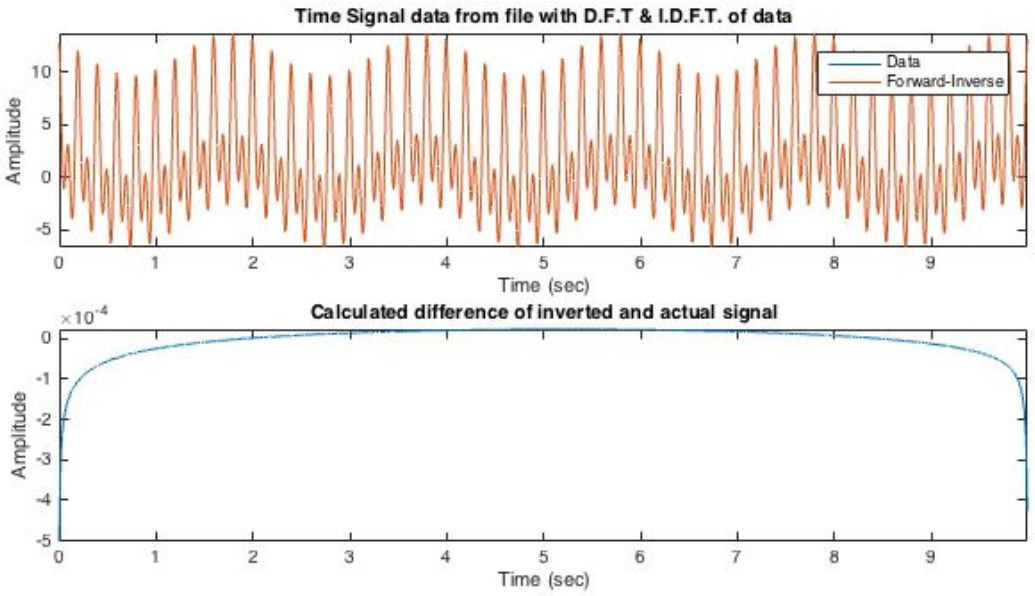
1. See Appendix I for MatLab code
2. See Figure Below



This figure contains the unshifted, unnormalized output of the DFT as well as the amplitude and phase spectra of the shifted and normalized DFT.

3. See Appendix I for MatLab code

4. See Figure Below



This figure contains a superimposed plot of the IDFT of the DFT of the data and the raw data itself. The second plot shows the difference between the two. This difference cannot be seen in the superimposed plot because they are so similar, on the order of 10^{-4}. This means that although they are not exactly the same, the two signals are essentially the same signal containing the same information.

Appendix I: The Code

```
function Lab05()
    clear;

    filename = 'data.dat';
    data = tblread(filename);
    time = data(:,1);
    value = data(:,2);
    forward = DFT(value, 'forward');
    forwardFixed = dftfix(forward, 'forward');
    inverseFixed = DFT(dftfix(forwardFixed, 'inverse'), 'inverse');

    figure('position', [0, 0, 700, 350]);
    orient tall;
    subplot(2,1,1);
    plot(time, value);
    hold on;
    plot(time, inverseFixed);
    axis tight;
    xlabel('Time (sec)');
    ylabel('Amplitude');
    title('Time Signal data from file with D.F.T & I.D.F.T. of data');
    legend('Data', 'Forward-Inverse');

    subplot(2,1,2);
    plot(time, value - inverseFixed);
    axis tight;
    xlabel('Time (sec)');
    ylabel('Amplitude');
    title('Calculated difference of inverted and actual signal');

    figure('position', [0, 0, 700, 500]);
    subplot(4,1,1);
    stem(real(forward));
    hold on;
    stem(imag(forward))
    axis tight;
    xlabel('Sample Number');
    ylabel('Amplitude');
    title('Raw D.F.T.');
    legend('Real', 'Imaginary');

    subplot(3,1,2);
    stem(Fk(time), abs(forwardFixed));
    axis tight;
    xlabel('Sample Number');
```

```
    ylabel('Amplitude');
    title('Shifted and Normalized D.F.T. Amplitude Spectrum');

    subplot(3,1,3);
    stem(Fk(time), angle(forwardFixed));
    axis tight;
    xlabel('Frequency (Hz)');
    ylabel('Phase (rad)');
    title('Shifted and Normalized D.F.T. Phase Spectrum');
end

function [result] = Fk(times)
    N = length(times);
    Ts = (times(N) - times(1)) / (N - 1);
    dF = 1 / (N * Ts);
    result = ((0:N-1) - ceil(N/2))*dF;
end

function [result] = dftfix(values, direction)
    N = length(values);
    Nyquist = ceil(N / 2);
    result = values;
    if strcmpi(direction, 'forward')
        result = result .* (1/N);
    end
    result = vertcat(result(Nyquist+1:N),result(1:Nyquist));
end

function [result] = DFT(values, direction)
    N = length(values);
    d = (strcmpi(direction, 'forward') * 2) - 1;
    matrix = zeros(length(values));
    Nyquist = ceil(N / 2);
%     if d == -1
%         % values = vertcat(values(Nyquist+1:N), values(1:Nyquist));
%         values = dftfix(values, 'backward');
%     end
    for n = 0:N-1
        for k = 0:n
            if n == 0
                operand = 1;
            else
                operand = exp(2i*3.1415926*n*k*d/N);
            end
            matrix(n+1,k+1) = operand;
            if n ~= k
                matrix(k+1,n+1) = operand;
            end
```

```
        end
    end
    result = matrix * values;
%     if d == 1
%         % result = result .* (1/N);
%         % result = vertcat(result(Nyquist+1:N),result(1:Nyquist));
%         result = dftfix(result, 'forward');
%     end
end
```