TSRT78 - Homework 1

axeso712

November 2023

1 Question

We consider the following signal composing two sinusoids:

$$s(t) = \sin(t) + \sin(1.2t). \tag{1}$$

We cannot measure the signal directly. The measured signal is given by:

$$y(t) = s(t) + e(t), \tag{2}$$

where e is Gaussian noise of zero-mean and variance of 0.01. The signal $y(t)$ is now sampled N times with a sampling interval of $T_s = 1$ s. Please write Matlab code to generate $N = 30$ samples of the observed signal y and compute its frequency model via discrete Fourier transform. You only need to plot the magnitude of the model with proper frequency axis.

1. Please explain why it is not possible to distinguish the two peaks of the two sinusoids.

2. Is it possible to reveal the two peaks in frequency domain using the same set of 30 samples from the previous question? If so, please explain your solution and validate it in Matlab with a plot.

2 Solution

2.1 Problem with distinguishing the peaks

According to the Nyquist-Shannon sampling theorem, to accurately represent a continuous signal, it must be sampled at least at twice the highest frequency present in the signal (the Nyquist rate). This condition is met since the highest frequency component is $\frac{1.2}{2\pi} \approx 0.19$ and our sample time $T_s = 1$ is more than twice that. This means that we should be able to distinguish the peaks. However in the figure we cannot do so (Figure 2). That is because we are using too few samples for the DFT.

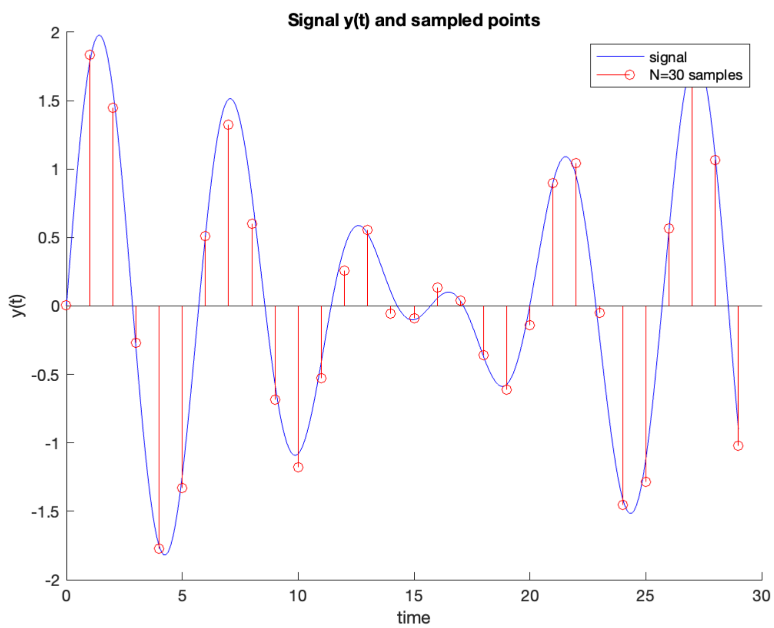

Figure 1: The signal and samples

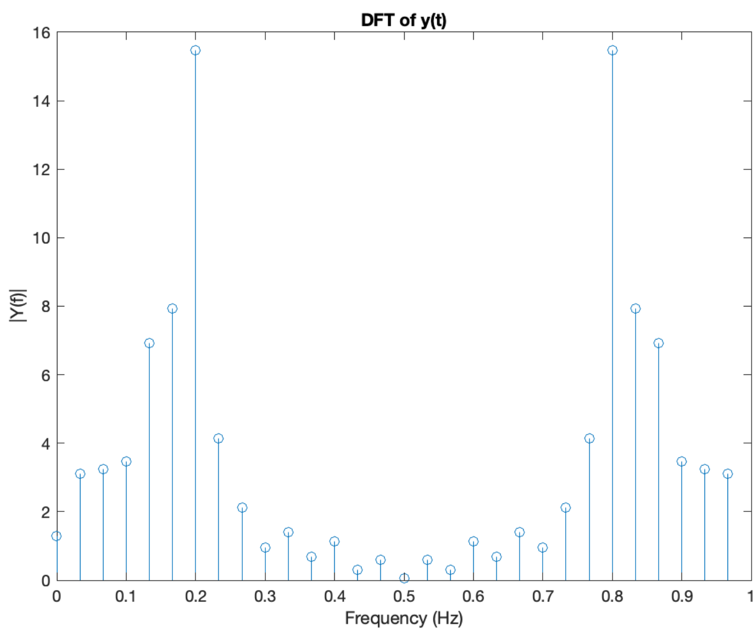

Figure 2: The DFT of the samples N=30

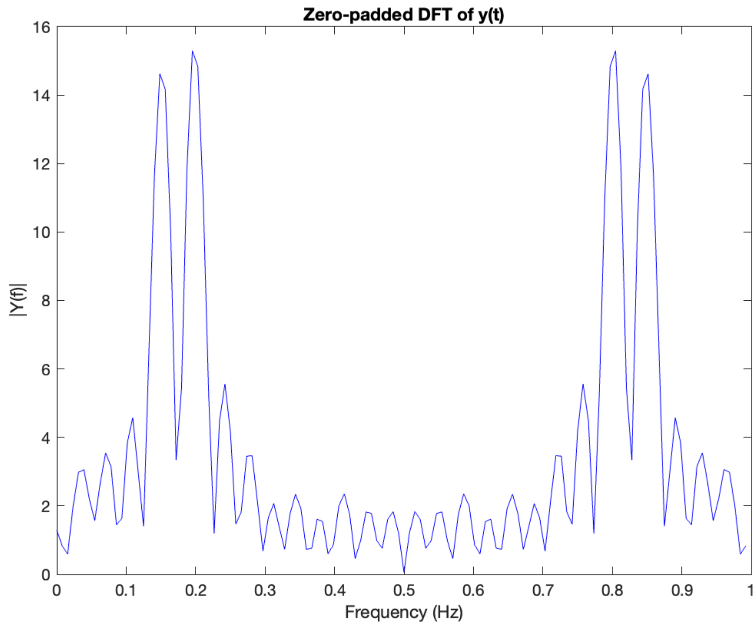

Figure 3: The zero-padded DFT of samples $N = 2^7$

2.2 How to reveal the peaks

To reveal the 2 peaks in the plot we use the technique of zero-padding for the DFT in order to more closely approximate the DTFT (Figure 4). This results in a much higher resolution. $2^7 = 128$ samples were used.

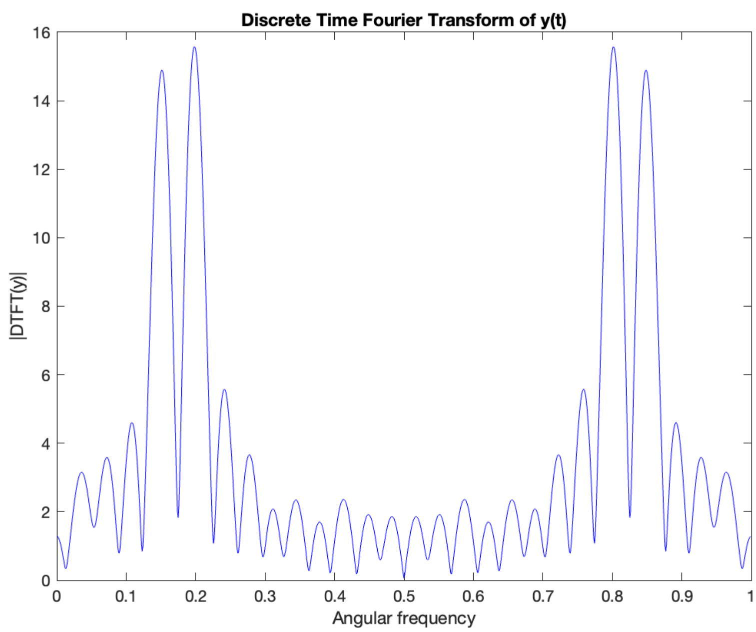

Figure 4: The DFTF of samples

3 Code

```
% Define parameters
N = 30;                       % Number of samples
Ts = 1;                       % Sampling interval in seconds
t = (0:N-1)*Ts;               % Time vector
var_e = 0.01;                 % Variance of Gaussian noise

% Generate signals
s = sin(t) + sin(1.2*t);      % Signal s(t)
e = sqrt(var_e)*randn(1,N);   % Gaussian noise e(t)
y = s + e;                    % Observed signal y(t)

figure;
hold on
tt = 0:0.1:N-1;
plot(tt, sin(tt) + sin(1.2*tt),'b')
stem(t,y,'r')
title('Signal y(t) and sampled points')
xlabel('time')
ylabel('y(t)')
legend('signal', 'N=30 samples')
print('./homework_1_signal', '-dpng')

% Perform the discrete Fourier transform (DFT)
Y = fft(y, N);

% Compute the frequency axis
f = (0:N-1)*(1/(Ts*N));

% Plot the magnitude of the DFT
figure;
stem(f, abs(Y));
title('DFT of y(t)');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
print('./homework_1_DFT', '-dpng')

new_N = 2^7;

% Zero-padding to improve frequency resolution
Y_zeropad = fft(y, new_N);

% Compute the new frequency axis for zero-padding
f_zeropad = (0:new_N-1)*(1/(Ts*new_N));
```

```
figure;
plot(f_zeropad, abs(Y_zeropad),'b');
title('Zero-padded DFT of y(t)');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
print('./homework_1_DFT_zero', '-dpng')

% DTFT
[Y_DTFT, w] = dtft(y);

figure;
plot(w,abs(Y_DTFT),'b')
xlabel('Angular frequency')
ylabel('|DTFT(y)|')
title('Discrete Time Fourier Transform of y(t)')
print('./homework_1_DTFT', '-dpng')
```