UNIVERSITY OF SHEFFIELD

Digital Signal Processing

Sampling, Analysis and Processing of the Sensor Data

AWABULLAH SYED

2/2/2021

Name: Awabullah M. Syed

Student Number: 2002 000 33

Module Code: ACS 6128

Module Assessor: Zi Qiang Lang

The MATLAB code can be found via the QR Code

Contents

Awab Syed

DSP Assignment

1.1. Time History & DFT Analysis of the Simulated Sensor Signal

Sensor data was extracted from the ‘sensor_data_for_DSP_assignment.mat’ file with the sampling rate of 1000Hz and was processed via MATLAB to deduce the time history and performed DFT analysis. Figure 1

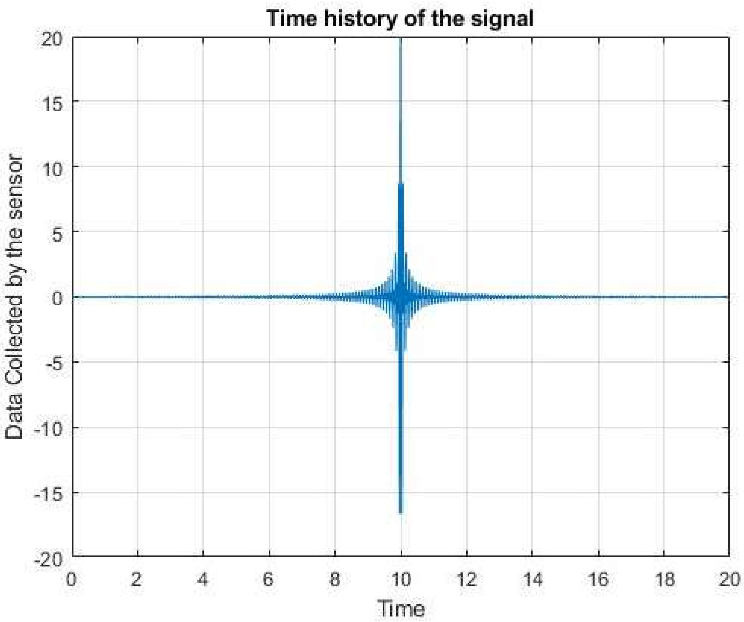


Figure 1 - Time history of the Signal with the sampling rate of 1000Hz

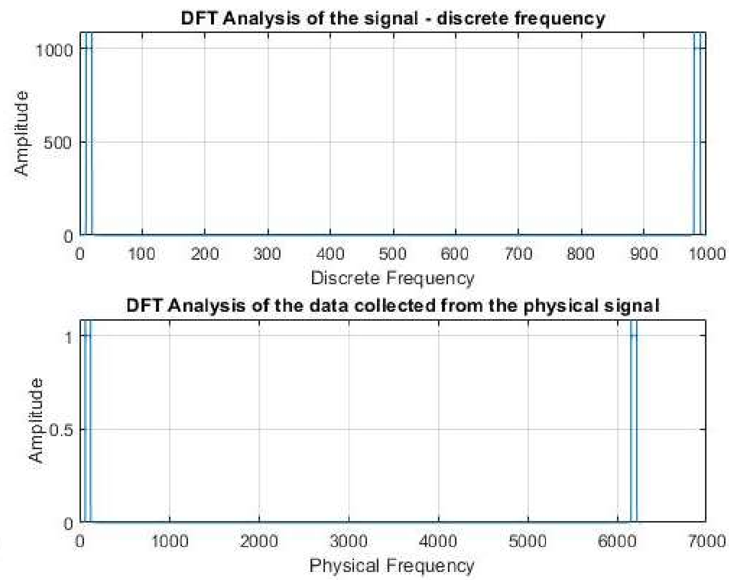
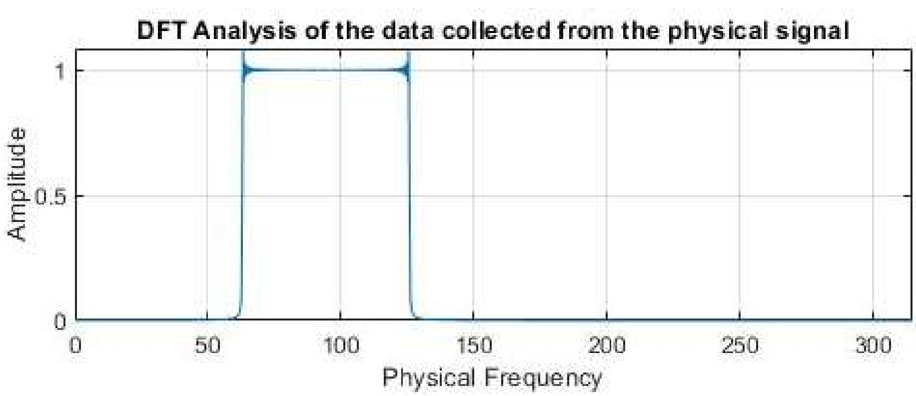


Figure 2 - DFT analysis of the signal

DFT shown in Figure 2 was obtained by implementing Fast Fourier Transform in MATLAB. The discrete frequency and the amplitude were appropriately scaled using Equation 1 and Equation 2, respectively to determine the Fourier amplitude as one and the physical frequency at around 6000Hz. Figure 3 illustrates the Fourier amplitude and physical frequency of the signal.

$$\omega = \frac{\Omega}{T} \qquad Eq.1$$

$$|X_c(jw)| = T\left|X(e^{j\Omega})\right| \qquad Eq.2$$



$$Amplitude = 1$$

$$Physical\ Frequency \approx 6000$$

Figure 3 - Physical Frequency and Fourier Amplitude after scaling applied

1.2. Down Sampling

The time history and the DFT Analysis of the down-sampled signal under the rates of 25Hz, 40Hz, and 50Hz was carried out via MATLAB. The Nyquist frequency shown in Equation 3 was used to analyze the DFT plots.

$$\omega_N = \frac{\omega_s}{2} < \omega_c \qquad Eq.3$$

Down-sampling of the initial signal and the sampling time was carried out using Equation 4 and Equation 5, respectively, which was then utilized in MATLAB to generate 'Time History' and DFT plot of the down-sampled signal.

$$Down\ sampling\ rate = \frac{Sampling\ rate}{new\ sampling\ rate} \qquad Eq.4$$

$25Hz = 40$; $40Hz = 25$; $50Hz = 20$

$$T_s = \frac{1}{Down - sampled\ rate} \qquad Eq.5$$

1.2.1. Time History of the signal down-sampled

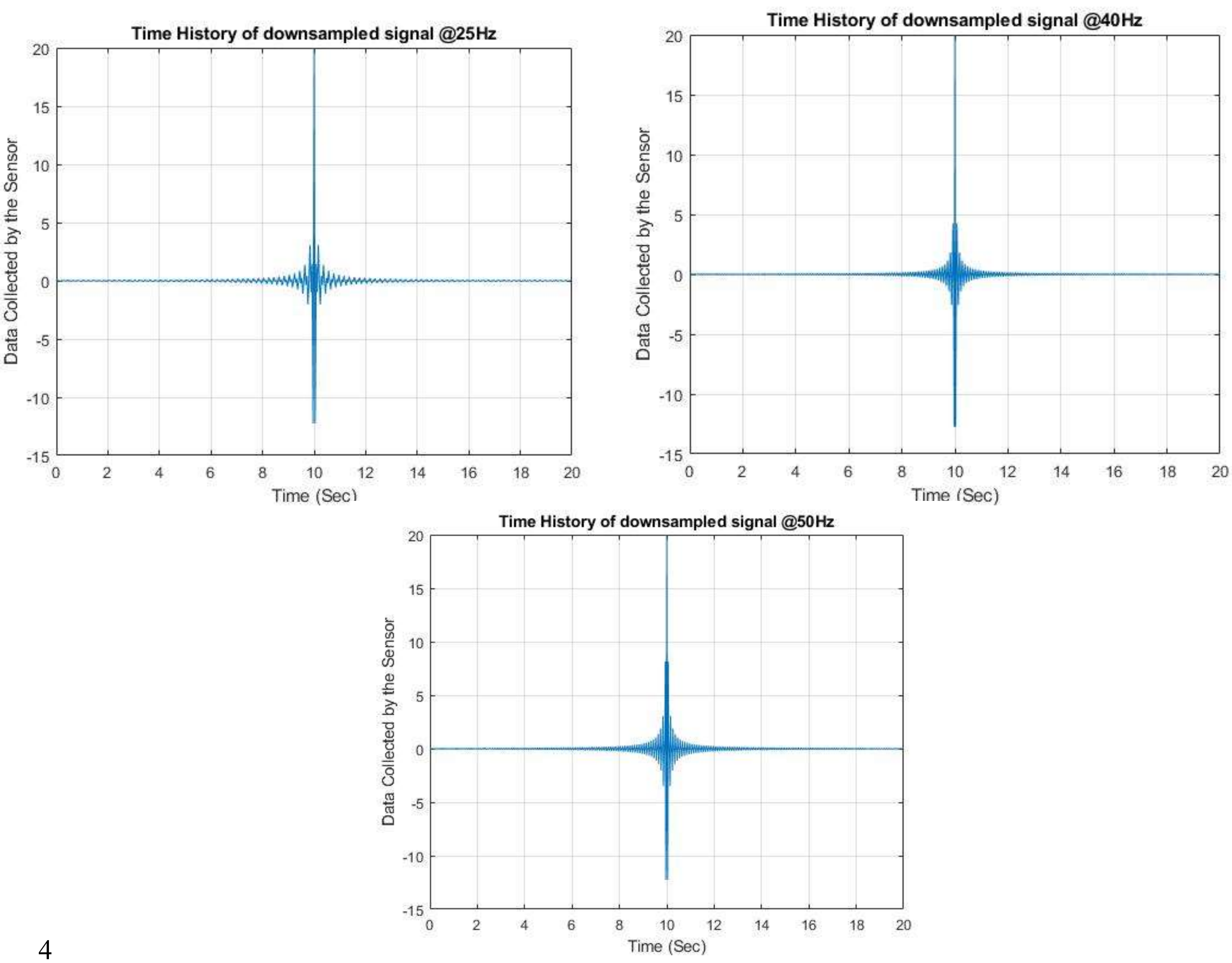


Figure 4 - Time History of the down-sampled signal under the rates of 25Hz, 40Hz, and 50Hz

1.2.2. DFT of the signal down-sampled

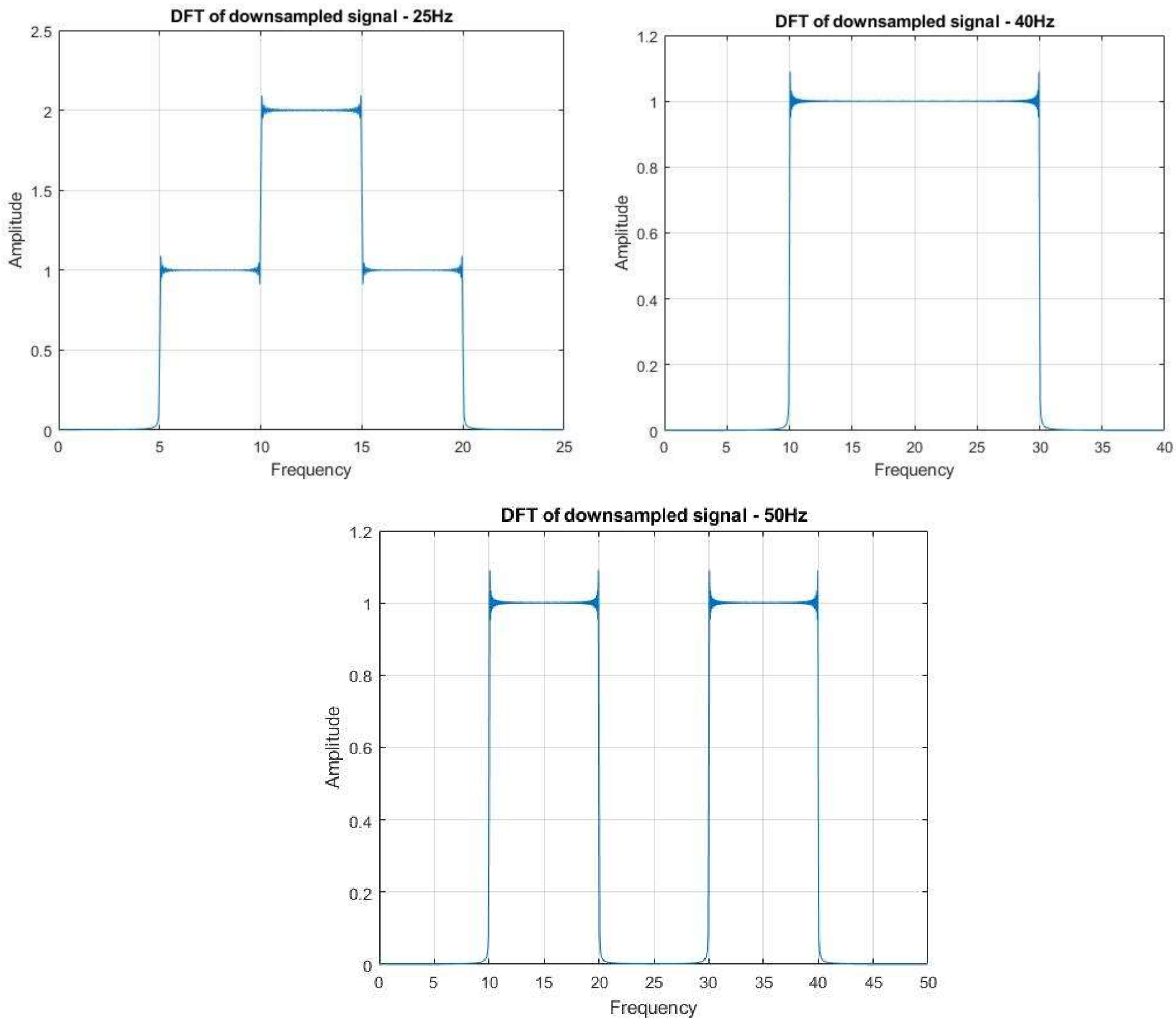


Figure 5 - DFT Analysis of the down-smapled signal under the rates of 25Hz, 40Hz and 50Hz

The Nyquist frequency theorem is shown in Equation 3 was utilized and the Nyquist frequency was calculated and is shown below to determine whether the sampled data can represent the original signal.

$$\omega_N\ (Rate: 25Hz) = \frac{2\pi \times 40}{2} = 40\pi \qquad T = 0.04\ sec$$

$$\omega_N\ (Rate: 40Hz) = \frac{2\pi \times 25}{2} = 25\pi \qquad T = 0.025\ sec$$

$$\omega_N\ (Rate: 50Hz) = \frac{2\pi \times 20}{2} = 20\pi \qquad T = 0.02\ sec$$

Considering the DFT plots and the Nyquist frequency, it was deduced that the signal down-sampled under the rate of 50Hz was able to represent the original signal as clearly evident from Figure 5.

1.3. Relationship between the sensor signal and physical signal

The relationship between the sensor signal and the physical signal is given by Equation 6

$$u(t) = \sqrt{y(t)} \qquad Eq. 6$$

Time history and the DFT analysis of the signal was extracted from MATLAB which was processed further to determine the amplitude and the frequency range.

1.3.1. Time History of y(t)

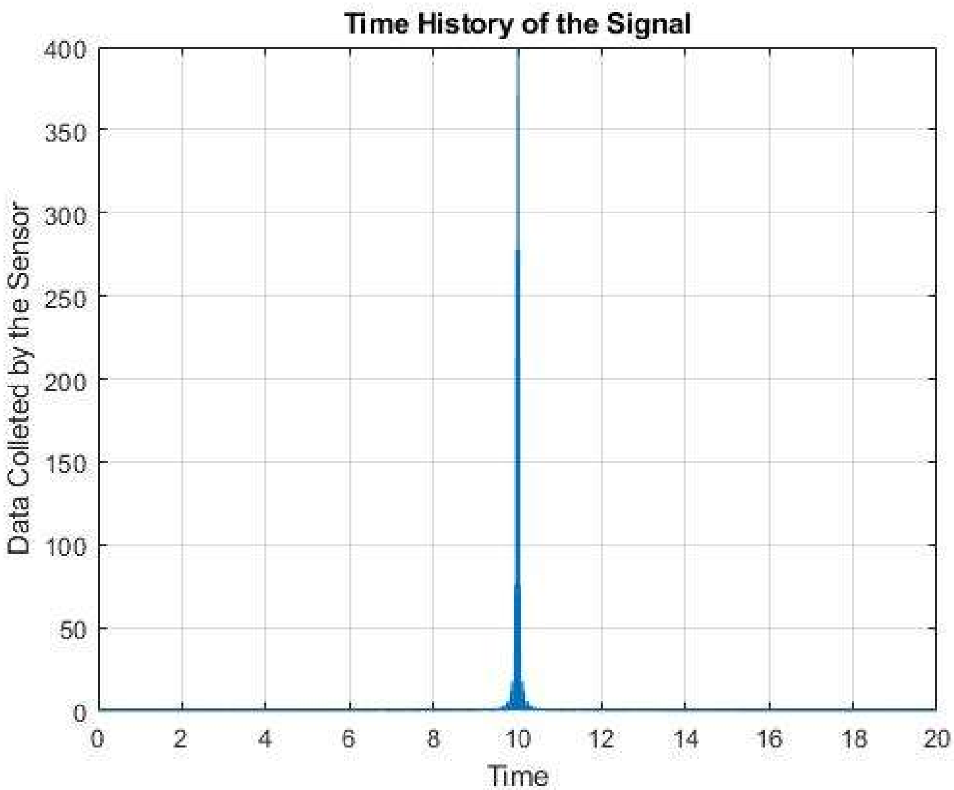


Figure 6 - Time History of y(t) signal

NOTE: The relevant code used to plot the time history is shown in the MATLAB Code section.

1.3.2. DFT Analysis

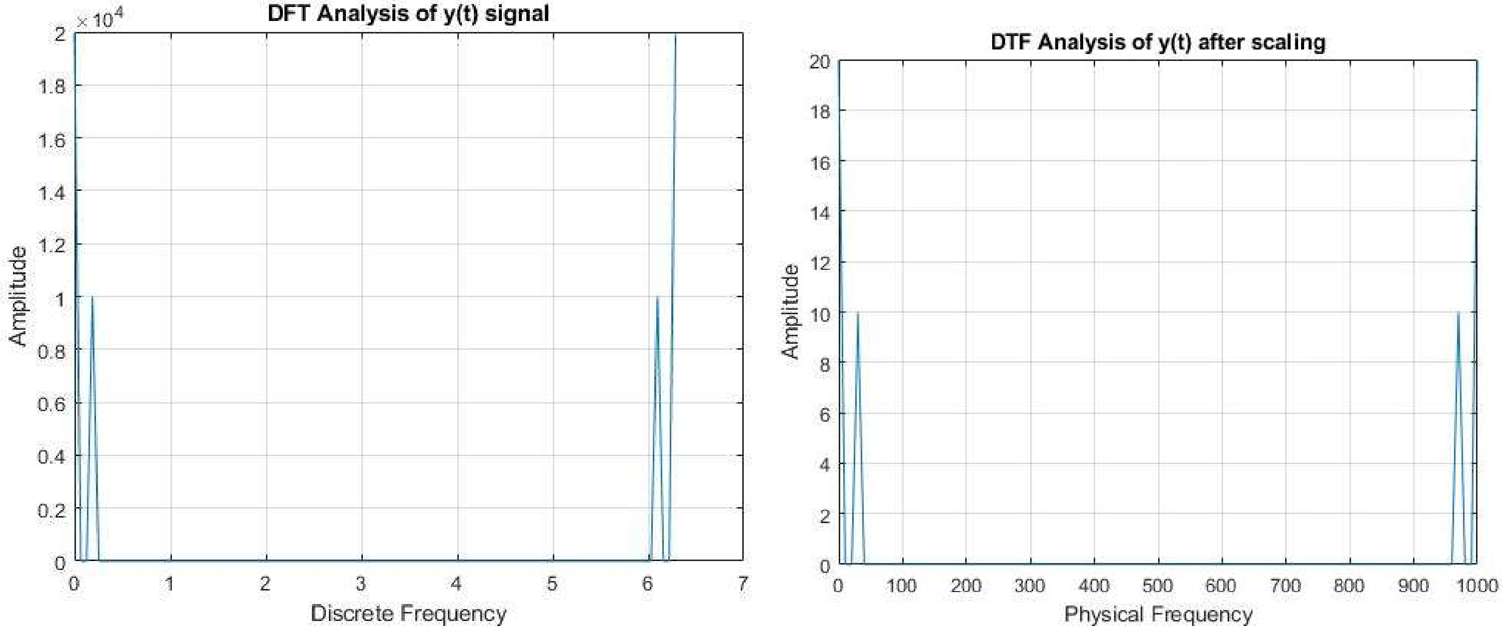


Figure 7 - DFT Analysis of the y(t) signal. Relevant code is shown in the MATLAB Code of the report

After scaling the DFT analysis, the amplitude of 10 and the maximum amplitude of 20 was determined over the frequency range of 0 – 1000.

Awab Syed

1.4. Low Pass Digital Filter

A digital filter was designed with the cut-off frequency, ω_1 of 10Hz with $\omega_2 = 19Hz$ and $\delta_2 = 0.1$.

The following steps were carried out to obtain the digital filter:

Step 1 – Design objectives were specified

$\omega_1 = tan\left(\frac{2\pi * 10 * 0.001}{2}\right) = 0.0314$

$\omega_2 = tan\left(\frac{2 * 19 * 0.001}{2}\right) = 0.0598$

$$\overline{\omega_1} = \frac{\omega_1}{\omega_1} = 1$$

$$\overline{\omega_2} = \frac{\omega_1}{\omega_2} = 1.9$$

$$|H_c(jw)|^2 = \ {\delta_2}^2 = \frac{1}{1+\omega^{2N}}$$

Step 2 – Calculating N, rounded to the nearest integer

$$N = \left\lceil \frac{log_{10}\left(\frac{1}{{\delta_2}^2} - 1\right)}{2log_{10}\omega^2} \right\rceil \qquad Eq.7$$

$$N = \left\lceil \frac{log_{10}\left(\frac{1}{0.1^2} - 1\right)}{2log_{10}19^2} \right\rceil = 1$$

Step 3 – Analogue filter as the transfer function $H_c(s)$

$$S_k = e^{jk\pi/N} \qquad Since\ N\ is\ odd$$

Poles on the left-hand side of the s-plane were considered due to stability.

$$H_c(s') = \frac{1}{s+1} = \frac{a_i}{s-p_i} = \frac{10}{s+10} \qquad Eq.8$$

Step 4 – Analogue filter to digital filter

Impulse Invariance Approach/Principle shown in Equation 8 was used to transfer the analogue filter to digital filter with the sampling time, $T = 0.001\ sec$

$$H(z) = \sum_{i=0}^{N} \frac{\alpha_i T}{1 - e^{P_{iT}} z^{-1}} \qquad Eq.9$$

$$H(z) = \frac{0.01}{1 - e^{-0.01} z^{-1}} \qquad T = 0.001 \text{ sec} \qquad Eq.10$$

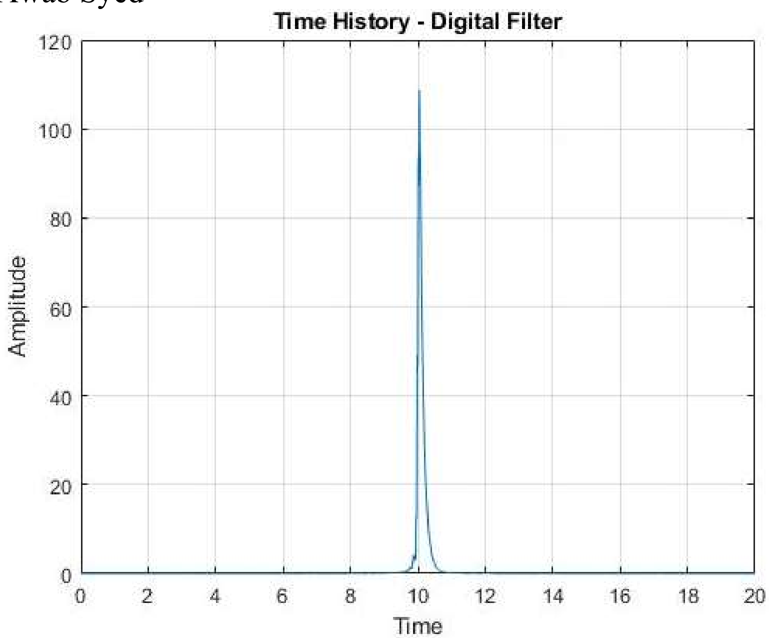


Figure 8 - Time History of the signal, y(t) with digital filter

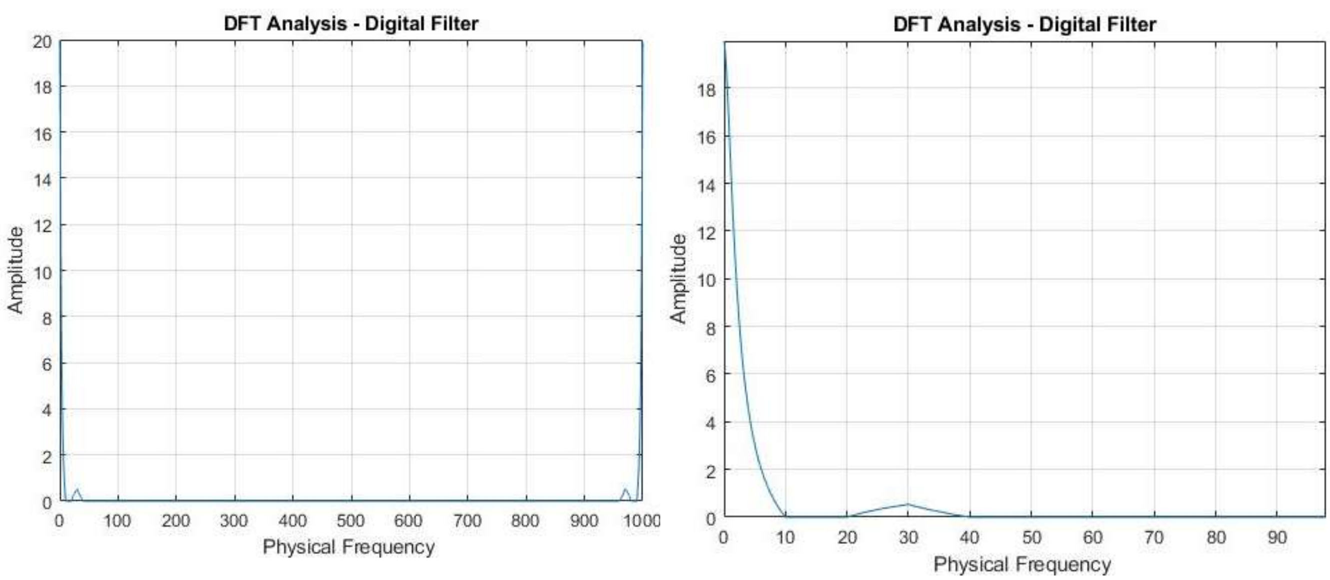


Figure 9 - DFT plot of the signal, y(t) with the digital filter implemented with T = 0.001 sec

Figure 9 shows the DFT plot of the signal, y(t) with the digital filter implemented with sampling time, $T_s = 0.001\ sec$. It is apparent from the zoomed-in DFT plot on the right side of Figure 9 that the physical signal, y(t0 is only meaningful over the frequency range of [0,10]Hz. However, the time history plot shown in Figure 8 suggests that the digital filter implemented can further be improved by utilizing the trial and error approach to determine the best ω_2 and δ_2 to achieve the ideal amplitude however, the trial and error approach was not employed to determine the optimal digital filter and can form part of further research.

Equation 9 shows the digital filter obtained via the Impulse Invariance Approach which was then implemented in MATLAB to obtain the time history and the DFT analysis shown in Figure 8 and Figure 9, respectively. Bilinear Transform Approach could also be used to transform the analogue filter shown in Equation 8 to a digital one however, in this case, the Impulse invariance approach was used and the comparison between various approaches is deemed outside the scope of this report. The author, (Leis) of 'Digital Signal Processing Using MATLAB for Students and Researchers' suggests that using Bilinear Transform Approach will enable the frequency response, $\left|H(e^{j\Omega})\right|$ and $|H_c(j\varpi)|$ to perfectly overlap which means that all the design specification could be achieved/inherited by the digital filter. Another approach could also be to utilize the MATLAB function, '*c2d*' to transform/implement the analogue filter as a digital filter.

2. MATLAB Code

2.1. Time History & DFT Analysis of the Simulated Sensor Signal

```
% Author: Awabullah Syed
% Date: 26 Jan 2021
% Module: Signal Processing
% NOTE: Please run the code section-wise to ensure necessary variables
%       needed in the second section is in the Workspace.
% DESCRIPTION: This code is broken down into 4 sections/points:
%             Point 1 - Performs the Time History and DFT analysis to the
%                       'Simulated sensor signal' - u
%             Point 2 - Downsampling of the sensor signal under the rates
%                       of 25 Hz, 40 Hz and 50 Hz.
%             Point 3 - Performs Time History and DFT analysis on y(t)
%                       signal
%             Point 4 - A low pass filter with the cut-off frequency of 10
%                       Hz
%-------------------------------------------------------------------------------------%
%-----------------------------Section: Point 1----------------------------------------%
load ('sensor_data_for_DSP_assignment.mat')  %Sensor signal
Fs = 1000;
T = 1/Fs;
N = 20000;
t = (0:N-1)*T; %Time
% DFT Analysis
yF = fft(u);
w = (0:2*pi/N:(N-1)*2*pi/N);  %vector with frequencies (omega)
capital_omega = w*T;
%------------------------------------Plot-------------------------------------------%
%------------------------------------Time History of the signal-------------------------%
figure(1)
plot(t,u)
grid on;
title('Time history of the signal')
xlabel('Time')
ylabel('Data Collected by the sensor')
%-------------------------------------- DT Plots ---------------------------------------%
figure(2)
subplot(2,1,1);
mag = abs(yF);
plot(w/(2*pi*T),mag);  %(2*pi*T) %capital omega
grid on
title ('DFT Analysis of the signal - discrete frequency');
xlabel ('Physical Frequency');
ylabel ('Amplitude')
subplot(2,1,2)
```

```
plot(w/T,(T*mag))  %w/T
title ('DFT Analysis of the data collected from the physical signal');
xlabel ('Angular Frequency')
ylabel ('Amplitude')
grid on
```

2.2. Down-Sampling

```
%------------------------------------------Section: Point 2-----------------------------%
%---------------------------------------------Down Sampling----------------------------%
t20 = [0:0.001:20]; % Used to generate Time history of the signal
%---------------------------------------Down Sample (25Hz)-------------------------------%
down_sample = 40; %1000/25 %1000/40 %1000/50
fs1 = 25; %25 Hz
T1 = 1/fs1;
N1 = length(u)/down_sample;
t1 = [0:(N1-1)*T1];
down_t = downsample(t1,down_sample);
y = downsample(u,down_sample); %downsampling the frequency
yF1 = fft(y);
w1 = (0:2*pi/N1:(N1-1)*2*pi/N1);  %vector with frequencies (omega)
t1_1 = downsample(t20,down_sample); %Downsampling Time (25 Hz)
%--------------------------------------Down Sample (40Hz)----------------------------------%
down_sample1 = 25; %1000/25 %1000/40 %1000/50
fs2 = 40; %25 Hz
T2 = 1/fs2;
N2 = length(u)/down_sample1;
t2 = [0:(N2-1)*T2];
y1 = downsample(u,down_sample1); %downsampling the frequency
yF2 = fft(y1);
w2 = (0:2*pi/N2:(N2-1)*2*pi/N2);  %vector with frequencies (omega)
t2_1 = downsample(t20,down_sample1); %Downsampling Time (40Hz)
%---------------------------------------Down Sample (50Hz)---------------------------------%
down_sample2 = 20; %1000/25 %1000/40 %1000/50
fs3 = 50; % Hz
T3 = 1/fs3;
N3 = length(u)/down_sample2;
t3 = [0:(N3-1)*T3];
y2 = downsample(u,down_sample2); %downsampling the frequency
yF3 = fft(y2);
w3 = (0:2*pi/N3:(N3-1)*2*pi/N3);  %vector with frequencies (omega)
t3_1 = downsample(t20,down_sample2); %Downsampling Time (50Hz)
%--------------------------------------DFT Down Sample Plots------------------------------%
figure (3)
subplot(3,1,1) %25 Hz
plot(w1/(2*pi*T1),(T1*abs(yF1))); grid on
xlabel ('Frequency')
```

```
ylabel('Amplitude')
title('DFT of downsampled signal - 25Hz')
subplot(3,1,2) %40 Hz
plot(w2/(2*pi*T2),(T2*abs(yF2))); grid on
xlabel ('Frequency')
ylabel('Amplitude')
title('DFT of downsampled signal - 40Hz')
subplot(3,1,3) %50 Hz
plot(w3/(2*pi*T3),(T3*abs(yF3))); grid on
xlabel ('Frequency')
ylabel('Amplitude')
title('DFT of downsampled signal - 50Hz')
%-----------------------------------Separate down sampling plots--------------------------%
figure (4)
plot(w1/(2*pi*T1),(T1*abs(yF1))); grid on
xlabel ('Frequency')
ylabel('Amplitude')
title('DFT of downsampled signal - 25Hz')
figure (5)
plot(w2/(2*pi*T2),(T2*abs(yF2))); grid on
xlabel ('Frequency')
ylabel('Amplitude')
title('DFT of downsampled signal - 40Hz')
figure (6)
plot(w3/(2*pi*T3),(T3*abs(yF3))); grid on
xlabel ('Frequency')
ylabel('Amplitude')
title('DFT of downsampled signal - 50Hz')
%--------------------------Time History of Each Down-Sampled Signal----------------------%
figure (7)  % Time History of downsampled Signal (25 Hz)
plot(t1_1(1:end-1),y)
title ('Time History of downsampled signal @25Hz')
xlabel('Time (Sec)')
ylabel('Data Collected by the Sensor'); grid on
figure (8)   % Time History of downsampled Signal (40 Hz)
plot(t2_1(1:end-1),y1)
title ('Time History of downsampled signal @40Hz')
xlabel('Time (Sec)')
ylabel('Data Collected by the Sensor'); grid on
figure (9) % Time History of downsampled Signal (40 Hz)
plot(t3_1(1:end-1),y2)
title ('Time History of downsampled signal @50Hz')
xlabel('Time (Sec)')
ylabel('Data Collected by the Sensor'); grid on
```

2.3. Relationship between the sensor signal and the physical signal

```
%% ------------------------------------------Point 3--------------------------------------%
%-----------------------Relationship between the sensor signal & physical signal----------%
y = u.^2;
figure (10) % Time History of y(t)
plot(t,y)
title ('Time History of the Signal')
xlabel('Time')
ylabel('Data Colleted by the Sensor') ; grid on
yf33 = fft(y);
w = [0:2*pi/N:(N-1)*2*pi/N];
figure(11)   %DFT Analysis Plot
plot(w,abs(yf33))
title('DFT Analysis of y(t) signal')
xlabel('Discrete Frequency')
ylabel('Amplitude'); grid on
figure (12)    %DFT Anaysis Plot after scaling
plot(w/(2*pi*T),T*abs(yf33))
xlabel('Physical Frequency')
title ('DTF Analysis of y(t) after scaling')
ylabel('Amplitude'); grid on
%plot (w/(2*pi*T),abs(yf))
```

2.4. Low-Pass Digital Filter

```
%-----------------------------------------Digital Filter-----------------------------------%
s = tf ('s');
Fs = 1000;
T = 1/Fs;
N = 20000;
t = (0:N-1)*T; %Time
num_sys = [0 0.01];
den_sys = [1 -0.99];
yfil = filter(num_sys,den_sys,y);
figure (13)
plot(t,yfil); grid on
xlabel('Time')
ylabel('Amplitude')
title('Time History - Digital Filter')
yf44 = fft(yfil); %DFT
w = [0:2*pi/N:(N-1)*2*pi/N];
figure(14)
plot(w/(2*pi*T),T*abs(yf44)); grid on
xlabel ('Physical Frequency ')
ylabel('Amplitude')
title('DFT Analysis - Digital Filter')
```

```
figure (15)
plot(w,abs(yf44))
xlabel ('Frequency ')
ylabel('Amplitude')
title('DFT Analysis - Digital Filter'); grid on
```

Bibliography

Leis, J. W. (n.d.). *Digital Signal Processing Using MATLAB for Students and Researchers* . Wiley .