Advanced Mathematics for Engineers

ABSTRACT

Signal processing consists of the analysis of signals by using mathematical theorems and software simulators, another important concept for engineers is the analysis of complex circuits by using Laplace Impedance Modelling.

INTRODUCTION

This report aims to explain some of the most essential topics in engineering by applying software simulators such as MATLAB and hand calculations, demonstrating the importance and procedure of some theorems such as Shannon's theorem and the application of Laplace transform in complex circuits.

Keywords

MATLAB, Continuous signal, Discrete signal, Nyquist, Shannon's theorem, Sampling, aliasing, Laplace, Impedance.

1. DISCRETE TIME AND ALIASING

1.1 Continuous Time Signal.

1.1.1 Time base.

A continuous time signal depends on the time. To produce a sequence of values in time, a time-base is required (*Equation 1*), using these values concerning time is possible to recreate a sinusoidal signal.

$$t = 0:0.001:2$$

Equation-1: Time-based.

1.1.2 Sinusoidal wave.

Using the time base was possible to represent the sinusoidal wave formed from sine and cosine waves with different frequencies (*Figure 1*). Combining the different signals (*Equation 2*), it was possible to recreate the continuous time signal in MATLAB (*Figure 2*).

$$X(t) = \sin(3t) - \cos(2t) + \sin(9t) + \cos(7t)$$

Equation-2: Sinusoidal wave.

Angular Frequency (ω)	Convert	Frequency (f)
3 rad/s	$\frac{\omega}{2\pi}$	0.4775 Hz
2 rad/s		0.3183 Hz
9 rad/s		1.4324 Hz
7 rad/s		1.1141 Hz

Figure-1: Frequencies table.

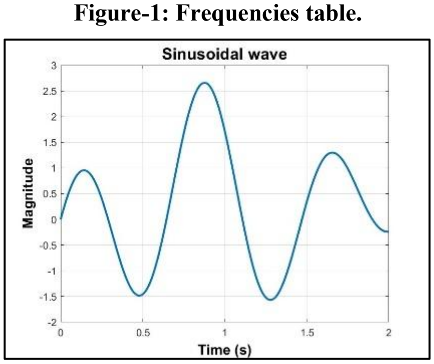


Figure-2: Sinusoidal wave

To convert this analogue signal into digital signals, the analogue signal must be sampled by using the highest frequency. This can be obtained by converting from radians to Hertz the sinusoidal equation.

1.2 Discrete Signal.

To avoid under-sampling and produce aliasing, it is important to follow Shannon's theorem (*Figure 3*)

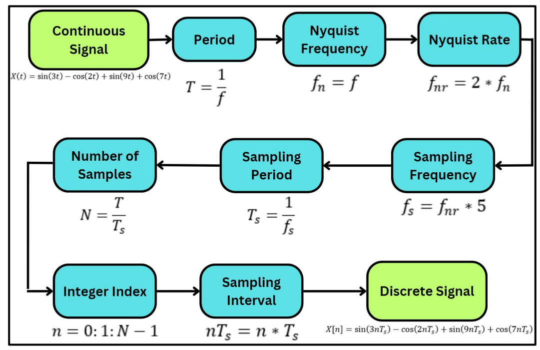


Figure-3: Procedure diagram.

Using a software simulator such as MATLAB, it is possible to represent and calculate the samples and recreate the discrete signal (*Figure 4*).

```
%% Discrete Signal Workout
T=1/f3; %% Period
fn=f3; %% Nyquist Frequency
fnr=2*fn; %% Nyquist Rate
fs=fnr*5; %%Sampling Frequency
Ts=1/fs; %% Sampling period
N=T/Ts; %% Number of Samples
n=0:1:3*N-1; %% Integer Index
nTs=n*Ts; %% Sampling Interval
```

Figure-4: Discrete signal workout in MATLAB.

1.2.1 Period and Nyquist frequency.

Once the highest frequency was found, it was possible to obtain the sinusoidal wave period by calculating the frequency inverse (*Equation 3*).

$$T = \frac{1}{f}$$
$$T = \frac{1}{1.4324}$$
$$T = 0.6981$$

Equation-3: period.

```
T=1/f3; %% Period
T =

    0.6981
```

Figure-5: Period code in MATLAB.

Another important parameter is the Nyquist frequency for this task, the Nyquist frequency will be the highest frequency in the analogue signal (*Equation 4*).

$$f_n = f$$
$$1.4324 = 1.4324$$

Equation-4: Nyquist frequency.

```
fn=f3; %% Nyquist Frequency
fn =

    1.4324
```

Figure-6: Nyquist frequency code in MATLAB.

1.2.2 Nyquist rate and sampling frequency.

Nyquist rate is the minimum sampling rate at which a signal can be sampled and still being able to be reconstructed without being affected by distortions (*Equation 5*). This parameter is important to avoid over-sampled or under-sampled signals [1].

$$f_{nr} = 2 * f_n$$
$$f_{nr} = 2 * 1.4324$$
$$f_{nr} = 2.8648$$

Equation-5: Nyquist rate.

```
fnr=2*fn; %% Nyquist Rate
fnr =

    2.8648
```

Figure-7: Nyquist rate code in MATLAB.

In theory, a signal can be reconstructed from its samples by using a sampling frequency more than twice the Nyquist rate [2] for this signal, the Nyquist rate was multiplied by five (*Equation 6*).

$$f_s = f_{nr} * 5$$
$$f_s = 2.8648 * 5$$
$$f_s = 14.3239$$

Equation-6: Sampling frequency.

```
fs=fnr*5; %%Sampling Frequency
fs =

   14.3239
```

Figure-8: Sampling frequency code in MATLAB.

1.2.3 Sampling period and the number of samples.

With the sampling frequency was possible to obtain the sampling period by calculating the inverse of the sampling frequency. This value represents the difference between two consecutive samples (*Equation 7*).

$$T_s = \frac{1}{f_s}$$
$$T_s = \frac{1}{14.3239}$$
$$T_s = 0.0698$$

Equation-7: Sampling period.

```
Ts=1/fs; %% Sampling period
Ts =

    0.0698
```

Figure-9: Sampling period code in MATLAB.

The number of samples represents the minimum number of samples required to reconstruct the signal; this value is calculated by dividing the period over the sampling period (*Equation 8*).

$$N = \frac{T}{T_s}$$
$$N = \frac{0.6981}{0.0698}$$
$$N = 10$$

Equation 8: Number of samples.

```
N=T/Ts; %% Number of Samples
N =

    10
```

Figure-10: Number of samples in MATLAB.

1.2.4 Integer index and sampling interval.

The integer index was represented in MATLAB (*Figure 11*) to display the number of samples. The number of samples is a whole number, and the increment applied in the code is a whole number (*Equation 9*).

```
n=0:1:3*N-1; %% Integer Index
```

Figure-11: Integer index code.

$$n = 0{:}1{:}N-1$$
$$n = 0{:}1{:}10-1$$
$$n = 0{:}1{:}9$$

Equation-9: Integer index.

Using the integer index was possible to obtain the value of each sample respect in time by multiplying the sampling period by the integer index (*Equation 10*).

$$nT_s = n * T_s$$

Equation-10: Sampling interval.

1.2.5 Result.

Once all the values have been found, the discrete signal was obtained (*Figure 12*), this signal is not continuous, and it is represented by the samples obtained applying Shannon's Theorem (*Equation 11*).

$$X[n] = \sin(3nT_s) - \cos(2nT_s) + \sin(9nT_s) + \cos(7nT_s)$$

Equation-11: Discrete signal.

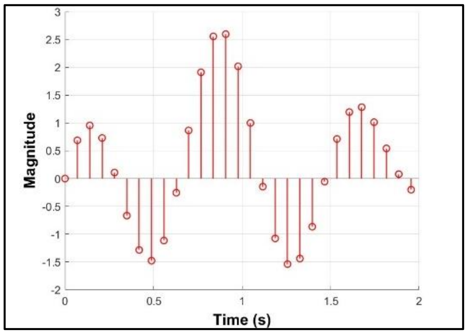


Figure-12: Discrete Signal.

If the two waves are plotted in the same graph, it is possible to observe how the samples of the discrete signal match the continuous time signal (*Figure 13*).

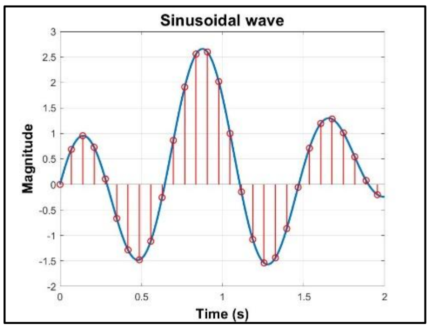


Figure-13: Discrete and continuous signal.

1.3 Verification

To verify that the values obtained using MATLAB, the values from the discrete signal have been obtained by hand calculations, obtaining similar results. To keep the similarity, a minimum of four decimals have been used for the result.

N° of sample	Sample Period	X Coordinate
0	0.0698	0
1	0.0698	0.0698
2	0.0698	0.1396
3	0.0698	0.2094
4	0.0698	0.2792
5	0.0698	0.349
6	0.0698	0.4188
7	0.0698	0.4886
8	0.0698	0.5584
9	0.0698	0.6282
10	0.0698	0.698
11	0.0698	0.7678
12	0.0698	0.8376
13	0.0698	0.9074
14	0.0698	0.9772

Figure-14: Hand calculations.

	1
1	0
2	0.0698
3	0.1396
4	0.2094
5	0.2793
6	0.3491
7	0.4189
8	0.4887
9	0.5585
10	0.6283
11	0.6981
12	0.7679
13	0.8378
14	0.9076
15	0.9774

Figure-15: MATLAB results.

1.4 Aliasing.

If Shannon's Theorem has not been applied, a phenomenon called aliasing can occur. This phenomenon is produced by a low sampling rate, this can be caused for noise contained inside the signal that is going to be reconstructed. This noise can contain higher frequencies that will affect the sampling, this means that if the sampling rate is not at least twice the higher frequency at the moment of sampling the signal, the result will produce an alias. The alias is the representation of an under-sampling signal [2] this can cause the loss of information, distortion in the signal obtained (*Figure 17*), and a false frequency.

Aliasing can be prevented by using Shannon's theorem; however, the signal may contain noise. To avoid aliasing, it is possible to increase the Nyquist rate by a factor of five, another possibility is to use a low pass filter circuit to block the frequencies above the Nyquist frequency. With these higher frequencies removed, it is possible to sample the signal without creating an alias of the original signal [3].

To prove the phenomenon of aliasing, the same signal was sampled without following Shannon's Theorem, for example, the frequency used to sample the discrete signal was not the highest frequency, and the Nyquist rate was not twice the highest frequency, this produces a different discrete signal (*Figure 16*) and an alias (*Figure 17).*

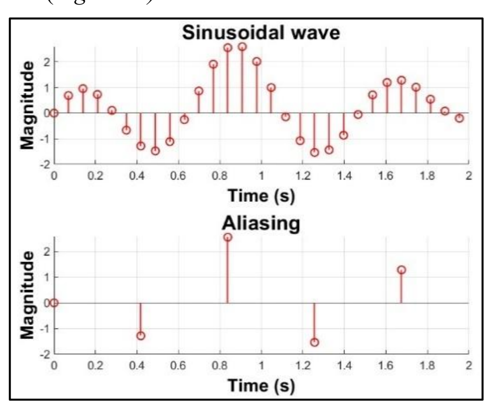


Figure-16: Aliasing

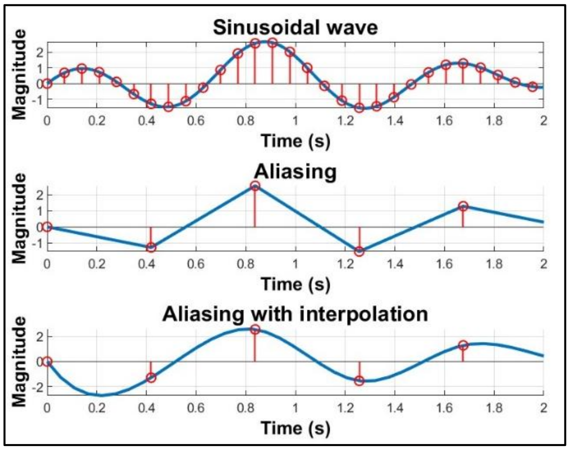


Figure-17: Alias produced by aliasing.

To obtain the frequency that produces the aliasing is required to subtract the original frequency minus the frequency used for the sampling [4] once the frequency is obtained, it is possible to obtain the period.

2. LAPLACE TRANSFORMS AND TRANSIENT ANALYSIS

2.1 Laplace Impedance Modelling

Laplace transform can be used to solve and analyse different circuits. This method converts circuits in the time domain into complex frequency domain circuits. This will allow working easily with components with a nonzero condition at time equals zero [5] , to prove this, a circuit has been designed in the time domain (*Figure 18*).

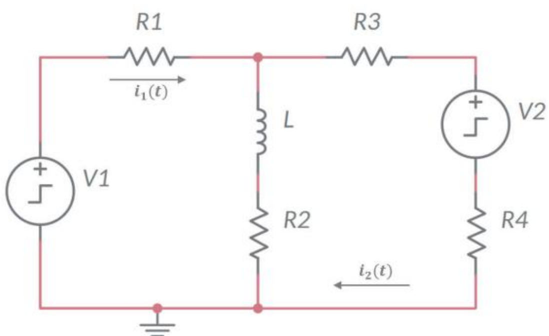


Figure-18: Circuit.

To analyse a circuit with an initial condition, it is necessary to convert all the components in the circuit into the Laplace domain, for example, the resistor (*Equation 12*), the inductor with the initial condition (*Equation 13*) and the voltage sources (*Equation 14*)

$$V_R(t) = R * i_R(t)$$
$$V_R(s) = R * i_R(s)$$

Equation-12: Laplace transform of a resistor.

$$V_L(t) = L\frac{di_L}{dt}$$
$$V_L(s) = sLI(s) - Li(0^-)$$

Equation 13: Laplace transform of an inductor.

$$V(t):\ \frac{V}{s}$$

Equation-14: Time domain to Laplace domain.

Once all the components have been converted into the Laplace domain, a new component is added to the circuit (*Figure 19*), this component is a voltage source, and it was obtained from the transfer function of the inductor [6].

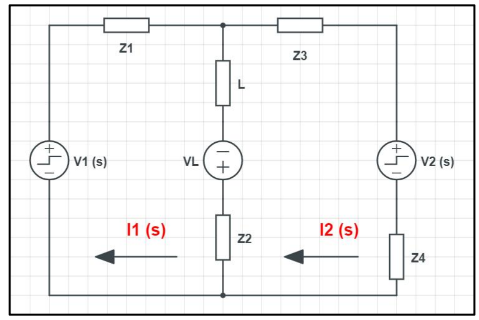


Figure-19: Circuit in the frequency domain.

2.2 Current equations

When circuits become more complicated, different types of methods can be used, for example, mesh analysis can be used to analyse this circuit, this method is based on Kirchhoff's laws and can be used to solve complex circuits by solving the simultaneous equation (*Equation 16*) with the branch currents [7].

$$V_1 + V_L = (Z_1 + Z_2 + L)I_1 - (Z_2 + L)I_2$$
$$V_2 + V_L = (Z_2 + L)I_1 - (Z_2 + Z_3 + Z_4 + L)I_2$$

Equation-15: Current equations.

Once the simultaneous equations for each current have been obtained, it is possible to convert them into a matrix and solve them using Crammer's rule (*Equation 17*).

$$\begin{bmatrix} Z_1 + Z_2 + L & -Z_2 - L & V_1 + V_L \\ L + Z_2 & -Z_2 - Z_3 - Z_4 - L & V_2 + V_L \end{bmatrix}$$

Equation-16: Matrix equation.

The matrix determinant is obtained (*Equation 18*).

$$-Z_1(Z_2+Z_3 + Z_4 + L) - L(Z_4+Z_3) - Z_2(Z_4+Z_3)$$

Equation-17: Determinant.

To solve it for each current, it is required to obtain the determinant of current one and current two, then these values will be divided by the initial determinant obtaining the symbolic equation for current one and current two.

$$\begin{vmatrix} V_1 + V_L & -Z_2 - L \\ V_2 + V_L & -Z_2 - Z_3 - Z_4 - L \end{vmatrix}$$

$$-V_1(Z_2+Z_3 + Z_4 + L) - V_L(Z_4+Z_3) - V_2(L+Z_2)$$

Equation-18: Current one determinant.

$$\begin{vmatrix} Z_1 + Z_2 + L & V_1 + V_L \\ L + Z_2 & V_2 + V_L \end{vmatrix}$$

$$-Z_1(V_2+V_L) - V_2(L+Z_2) - V_1(L+Z_2)$$

Equation-19: Current two determinant.

$$I_1 = \frac{-V_1(Z_2+Z_3+Z_4+L) - V_L(Z_4+Z_3) - V_2(L+Z_2)}{-Z_1(Z_2+Z_3+Z_4+L) - L(Z_4+Z_3) - Z_2(Z_4+Z_3)}$$

$$I_2 = \frac{-Z_1(V_2+V_L) - V_2(L+Z_2) - V_1(L+Z_2)}{-Z_1(Z_2+Z_3+Z_4+L) - L(Z_4+Z_3) - Z_2(Z_4+Z_3)}$$

Equation-20: Current one and two symbolic equations.

2.3 MATLAB

A MATLAB code has been designed to allow the user to introduce data (*Figure 20*) and obtain the transfer response of the system produced by the value of the components used.

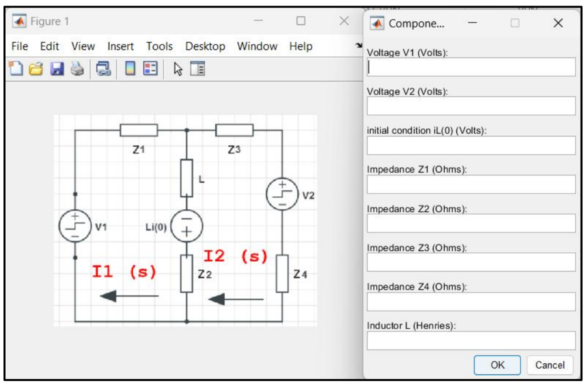


Figure 20: MATLAB dialogue box.

Components	Value
V_1	4V
V_2	3V
V_L	0.4V
L	0.3
R_1	33Ω
R_2	27Ω
R_3	75Ω
R_4	56Ω

Figure 21: Components.

The MATLAB code incorporates the symbolic equations for each current. Once the values of the component are introduced in MATLAB, the transfer function for current one and current two will be obtained.

$$I_1(s) = \frac{52.7s + 551}{49.2s^2 + 8751s}$$

$$I_2(s) = \frac{-12.9s - 72}{49.2s^2 + 8751}$$

Equation-21: Transfer function of current one and two.

Using the transfer function, it was possible to plot the impulse response of each current (*Figure 21*).

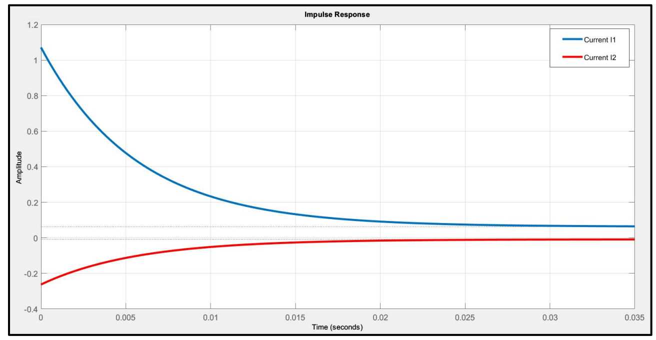


Figure-22: Laplace domain impulse response.

2.4 Inverse Laplace.

It is possible to obtain the continuous time domain equation for each current by applying the inverse Laplace transform, using the Laplace table, and applying the Partial Fraction Expansion techniques, the equation for current one and current two were obtained.

$$I_1(t)1.008e^{-177.87t} + 0.0629t$$

$$I_2(t) - 0.252e^{-177.87t} - 0.0082t$$

Equation-22: Inverse Laplace transform of current one.

Using these equations, it is possible to verify that the impulse response in the Laplace domain is the same impulse response obtained in the time domain (*Figure 23*).

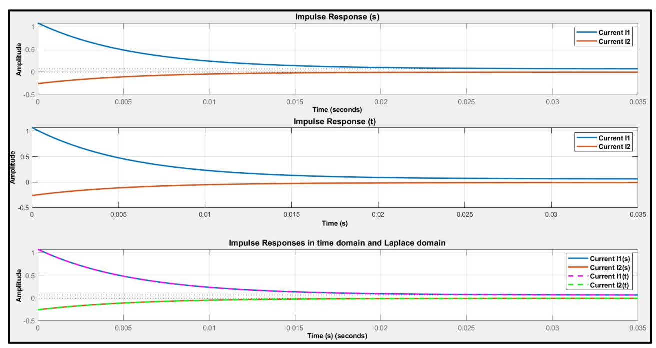


Figure-23: Laplace domain and Time domain impulse response.

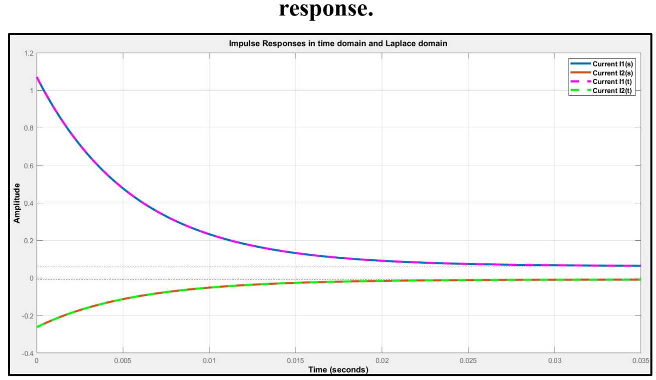


Figure 24: Comparison.

3. References

[1] M. K. Saini, "Tutorialspoint," 03 January 2022. [Online]. Available: https://www.tutorialspoint.com/what-is-nyquist-rate-and-nyquist-interval#. [Accessed 22 March 2023].

[2] A. R. Hambley, Electrical Engineering Principles and Applications, New Jersey: Pearson, 2011.

[3] AuthorCADENCE PCB SOLUTIONS, "Anti-aliasing Filter Design and Applications in Sampling," 2020. [Online]. Available: https://resources.pcb.cadence.com/blog/2020-anti-aliasing-filter-design-and-applications-in-sampling. [Accessed 22 March 2023].

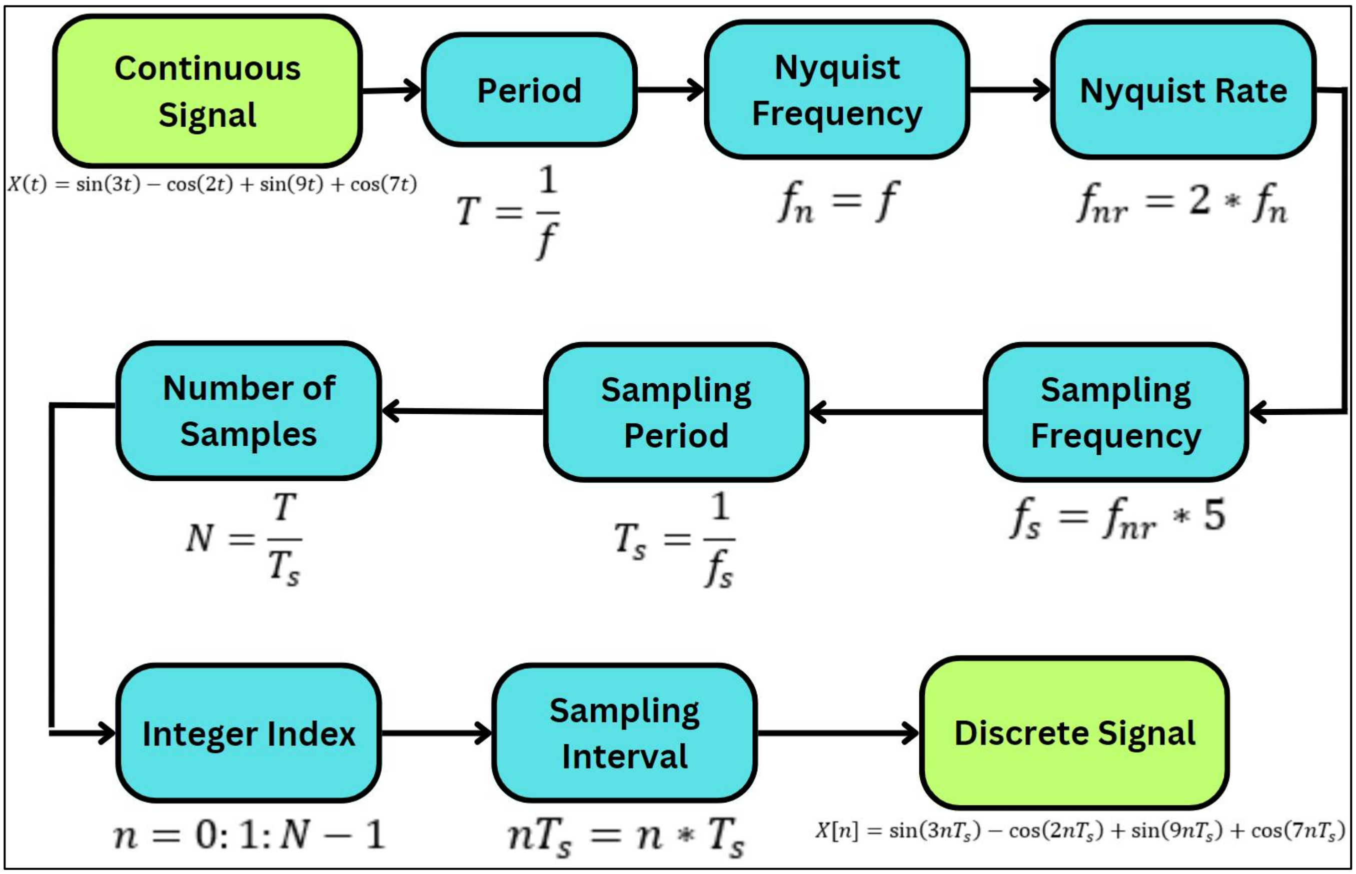
[4] M. Hasegawa-Johnson, "Sampling and Aliasing," Illinois, 2021.

[5] J. Semmlow, Signals and Systems for Bioengineers, Waltham, MA, 2012.

[6] S. Karris, Circuit Analysis II with MATLAB Computing and Simulink/SimPowerSystems Modeling, California: Orchard Publications, 2009.

[7] M. Plonus, Electronics and Communications for Scientists and Engineers, Illinois: Academic Press, 2001.

Appendix



```
%% DISCRETE TIME AND ALIASING
%% Time Base
t = 0:0.001:2;
%% Frequency values
f1=3/(2*pi);
f2=2/(2*pi);
f3=9/(2*pi);
f4=7/(2*pi);
%% Continuous Signal
X_t=sin(3*t)-cos(2*t)+sin(9*t)+cos(7*t);
%% Plot the graph
figure (1)
subplot(3,1,1) %% Subplot 1
plot(t,X_t,'LineWidth',2) %% Display graph
grid %% Grid command
hold on
%% Style of the Graph 1
ax=gca; % Command to change the axis without change the labels
ax.XLim=[0 2] % Limit values of axis Y
title('Sinusoidal wave','FontSize',14,'FontWeight','bold','color','black')
xlabel('Time (s)','FontSize',12,'FontWeight','bold','color','black')
ylabel('Magnitude','FontSize',12,'FontWeight','bold','color','black')
%% Discrete Signal Workout
T=1/f3; %% Period
fn=f3; %% Nyquist Frequency
fnr=2*fn; %% Nyquist Rate
fs=fnr*5; %%Sampling Frequency
Ts=1/fs; %% Sampling period
N=T/Ts; %% Number of Samples
n=0:1:3*N-1; %% Integer Index
nTs=n*Ts; %% Sampling Interval
%% Discrete Signals
Xn_1=sin(3*nTs);
Xn_2=cos(2*nTs);
Xn_3=sin(9*nTs);
Xn_4=cos(7*nTs);
Xn=Xn_1-Xn_2+Xn_3+Xn_4;
```

```
%% Plot Discrete Signal
subplot(3,1,1)
stem(nTs,Xn,'LineWidth',1,'color','r')
saveas(gcf,'graph1.jpg')
hold off
```

```
%% Discrete Signal Alias Workout
T_A=1/f1; %% Period
fn_A=f1; %% Nyquist Frequency
fnr_A=fn_A; %% Nyquist Rate
fs_A=fnr_A*5; %%Sampling Frequency
Ts_A=1/fs_A; %% Sampling period
N_A=T_A/Ts_A; %% Number of Samples
n_A=0:1:3*N_A-1; %% Integer Index
nTs_A=n_A*Ts_A; %% Sampling Interval
```

```
%% Discrete Signals Alias
Xn_1A=sin(3*nTs_A);
Xn_2A=cos(2*nTs_A);
Xn_3A=sin(9*nTs_A);
Xn_4A=cos(7*nTs_A);
Xn_A=Xn_1A-Xn_2A+Xn_3A+Xn_4A;
```

```
%% Plot the graph Alias with Interpolation
subplot(3,1,2) %%
grid %% Grid command
hold on
%X_A = linspace(min(nTs_A), max(nTs_A), 100);
%Y_A = spline(nTs_A, Xn_A, X_A);
plot(nTs_A,Xn_A,'LineWidth',2)
%%plot(X_A, Y_A, 'LineWidth',2)
%% Style of the Graph Alias
ax=gca; % Command to change the axis without change the labels
ax.XLim=[0 2] % Limit values of axis Y
title('Aliasing','FontSize',14,'FontWeight','bold','color','black')
xlabel('Time (s)','FontSize',12,'FontWeight','bold','color','black')
ylabel('Magnitude','FontSize',12,'FontWeight','bold','color','black')
%% Plot Discrete Signal Alias
subplot(3,1,2)
stem(nTs_A,Xn_A,'LineWidth',1,'color','r')
hold off
```

```
%% Alias number 2
%% Plot the graph Alias with Interpolation
subplot(3,1,3) %% Subplot of the Alias with interpolation
grid %% Grid command
hold on
%% Interpolation command
X_A = linspace(min(nTs_A), max(nTs_A), 100);
Y_A = spline(nTs_A, Xn_A, X_A);
plot(X_A, Y_A, 'LineWidth',2) %% plot the alias
stem(nTs_A,Xn_A,'LineWidth',1,'Color','r') %% plot the discrete signal
%% Style of Graph Alias
ax=gca; % Command to change the axis without change the labels
ax.XLim=[0 2] % Limit values of axis Y
title('Aliasing with interpolation','FontSize',14,'FontWeight','bold','color','black')
xlabel('Time (s)','FontSize',12,'FontWeight','bold','color','black')
ylabel('Magnitude','FontSize',12,'FontWeight','bold','color','black')
hold off
saveas(gcf,'graph.jpg') %% Save the graph
```

n	Ts	X
0	0.0698	0
1	0.0698	0.0698
2	0.0698	0.1396
3	0.0698	0.2094
4	0.0698	0.2792
5	0.0698	0.349
6	0.0698	0.4188
7	0.0698	0.4886
8	0.0698	0.5584
9	0.0698	0.6284
10	0.0698	0.698
11	0.0698	0.7678
12	0.0698	0.8376
13	0.0698	0.9074
14	0.0698	0.9772

```
%% TASK 2
%%  Display the image
f=figure('Position',[80 50 400 317]); %% Image coordinates
image=imread("circuit1.png"); %% Image of the circuit
imshow(image); %% Show the images
pause(4)
%% Dialogue box %%
dlgtitle = 'Component values';%% Name of the dialogue box
prompt={'Voltage V1 (Volts):','Voltage V2 (Volts):','initial condition VL(0
dims=[1 40]; %% Dimensions of the text box
answer = inputdlg(prompt,dlgtitle,dims); %% user data
%% Convert to numbers
V_1 = str2num(answer{1}); %% Value V1
V_2 = str2num(answer{2}); %% Value V2
VL = str2num(answer{3}); %% Value VL
Z1 = str2num(answer{4}); %% Value Z1
Z2 = str2num(answer{5}); %% Value Z2
Z3 = str2num(answer{6}); %% Value Z3
Z4 = str2num(answer{7}); %% Value Z4
ZL = str2num(answer{8}); %% Value ZL
```

```
%% Components
syms s; %% Matlab toobox
s=tf('s'); %% Convert s in transfer function
V1=V_1/s; %% V1 transfer function
V2=V_2/s; %% V2 transfer function
Z1=33; %% Value for R1
Z2=27; %% Value for R2
Z3=75; %% Value for R3
Z4=56; %% Value for R4
L=ZL*s; %% L transfer function
VL=0.4; %% VL transfer function
%% DETERMINANT
det=(Z1+Z2+L)*(-L-Z2-Z3-Z4)-(+L+Z2)*(-Z2-L); %% Determinant of the matrix
%% DETERMINANT I1
D_I1=(V1+VL)*(-L-Z2-Z3-Z4)-(VL+V2)*(-Z2-L) %% Determinant of the current I1
%% I1 TF
I1=D_I1/det; %% Current one value
%% DETERMINANT I2
D_I2=(Z1+Z2+L)*(VL+V2)-(L+Z2)*(V1+VL); %% Determinant of the current I2
%% I2 TF
I2=D_I2/det; %% Current two value
figure(2)
subplot(3,1,1) %% Subplot number one
impulseplot(I1) %% Impulse response I1
hold on
impulseplot(I2) %% Impulse response I2
%% Graph style
title('Impulse Response (s)','FontSize',12,'FontWeight','bold','color','bla
legend('Current I1','Current I2','FontSize',10,'FontWeight','bold')
xlabel('Time','FontSize',10,'FontWeight','bold','color','black')
ylabel('Amplitude','FontSize',10,'FontWeight','bold','color','black')
grid
hold off
```

```
%% Transient Response %%
t=0:0.0001:0.035; %% Time base for the time-domain
%% Current 1
I_1=(1.008*exp(-177.87*t))+0.0629; %% Current response in time domain
subplot(3,1,2) %% Subplot 2
plot(t,I_1) %% Plot of the I1(t)
hold on
%% Current 2
I_2=(-0.254*exp(-177.87*t))-(0.0082);%% Current response in time domain
subplot(3,1,2) %% Subplot 2
plot(t,I_2) %% Plot of the I2(t)
%% Graph style
title('Impulse Response (t)','FontSize',12,'FontWeight','bold','color','bla
legend('Current I1','Current I2','FontSize',10,'FontWeight','bold')
xlabel('Time (s)','FontSize',10,'FontWeight','bold','color','black')
ylabel('Amplitude','FontSize',10,'FontWeight','bold','color','black')
grid
hold off
%% time and laplace graph
subplot(3,1,3) %%Subplot 3
impulseplot(I1) %% Impulse response I1
hold on
impulseplot(I2) %% Impulse response I2
%% Graph style
plot(t,I_1,'LineStyle','--','Color','magenta','LineWidth',2) %% I1(t) Plot
plot(t,I_2,'LineStyle','--','Color','green','LineWidth',2) %% I2(t) Plot
title('Impulse Responses in time domain and Laplace domain','FontSize',12,'
legend('Current I1(s)','Current I2(s)','Current I1(t)','Current I2(t)','Fon
xlabel('Time','FontSize',12,'FontWeight','bold','color','black')
ylabel('Amplitude','FontSize',12,'FontWeight','bold','color','black')
grid
saveas(gcf,'both.jpg')
hold off
```

$$\left[\begin{array}{cc|c} Z_1+L+Z_2 & -L-Z_2 & V_1+iL \\ L+Z_2 & -Z_2-Z_3-Z_4-L & V_2+iL \end{array}\right]$$

Top → $(Z_1+L+Z_2)(-Z_2-Z_3-Z_4-L)$ →

$-Z_2Z_1-Z_1Z_3-Z_1Z_4-LZ_4-LZ_2-LZ_3-L^2-Z_2^2-Z_2Z_3-Z_2Z_4-Z_2L-Z_1L$

Bot → $(L+Z_2)(-L-Z_2)$ →

$-L^2-LZ_2-LZ_2-Z_2^2$

det → Top − Bot → $-Z_2Z_1-Z_1Z_3-Z_1Z_4-LZ_4-LZ_3-Z_2Z_3-Z_2Z_4-Z_1L$,

$$\begin{bmatrix} V_1+iL & -L-Z_2 \\ V_2+iL & -Z_2-Z_3-Z_4-L \end{bmatrix}$$

Top → $(V_1+iL)(-Z_2-Z_3-Z_4-L)$ →

$-V_1Z_2-V_1Z_3-V_1Z_4-V_1L-iLZ_2-iLZ_3-iLZ_4-iL$

Bot → $(-L-Z_2)(V_2+iL)$ →

$-LV_2-LiL-Z_2V_2-Z_2iL$

det → Tot − Bot → $-V_1Z_2-V_1Z_3-V_1Z_4-V_1L-iLZ_3-iLZ_4+LV_2+Z_2V_2$,

$$\begin{bmatrix} Z_1+L+Z_2 & V_1+iL \\ L+Z_2 & V_2+iL \end{bmatrix}$$

B

Top → $(Z_1+L+Z_2)(V_2+iL)$ →

$Z_1V_2+Z_1iL+LV_2+LiL+Z_2V_2+Z_2iL$

Bot → $(V_1+iL)(L+Z_2)$

$V_1L+V_1Z_2+iLL+iLZ_2$

det → Top − Bot → $Z_1V_2+Z_1iL+LV_2+Z_2V_2-V_1L-V_1Z_2$,

$$I_1 \rightarrow \frac{-V_1(Z_2+Z_3+Z_4+L)-iL(Z_3+Z_4)+V_2(L+Z_2)}{-Z_1(Z_2+Z_3+Z_4+L)-L(Z_4+Z_3)-Z_2(Z_3+Z_4)}$$

$$I_2 \rightarrow \frac{Z_1(V_2+iL)+V_2(L+Z_2)-V_1(L+Z_2)}{-Z_1(Z_2+Z_3+Z_4+L)-L(Z_4+Z_3)-Z_2(Z_3+Z_4)}$$

$$I_1 = \frac{52.7s + 551}{49.2s^2 + 8751s} \rightarrow I_1 = \frac{1.07s + 11.2}{s^2 + 177.87s} \rightarrow \frac{1.07s + 11.2}{s(s + 177.87)} = I_1$$

$$f(s) = \underbrace{\frac{1.07s}{s(s+177.87)}}_{\text{Part 1}} + \underbrace{\frac{11.2}{s(s+177.87}}_{\text{Part 2}}$$

Part 1

$$\mathcal{L}^{-1}\left(\frac{1.07s}{s(s+177.87)}\right)$$

$$\left(\frac{1.07}{s+177.87}\right)$$

$$1.07\left(\frac{1}{s+177.87}\right)$$

$$1.07\,\mathcal{L}^{-1}\left(\frac{1}{s+177.87}\right) \rightarrow f(t) = e^{-at} = \mathcal{L}(f(t)) = \frac{1}{s+a}$$

$$\underline{\underline{1.07 \cdot e^{-177.87t}}}$$

Part 2 "Descompose System" "Cover up theorem"

$$\mathcal{L}^{-1}\left(\frac{11.2}{s(s+177.87)}\right) \rightarrow \frac{A}{s} + \frac{B}{s+177.87}$$

solve for A

$$F(s) = \frac{B}{s+177.87}\Bigg|_{s\to 0} = \frac{11.2}{0+177.87} \rightarrow 0.0629$$

solve for B

$$F(s) = \frac{A}{s}\Bigg|_{s\to -177.87} = \frac{11.2}{-177.87} \rightarrow -0.063$$

$$\frac{A}{s} \rightarrow \frac{0.0692}{s} \rightarrow \frac{a}{s} \text{ so } f(t) = 0.0692(t)$$

$$\frac{B}{s+a} \rightarrow \frac{-0.063}{s+177.87} \rightarrow Be^{-at} \text{ so } f(t) = -0.063e^{-177.87t}$$

Solution Part 1 and Part 2

$$1.07e^{-177.87t} + 0.0692 - 0.063e^{-177.87t}$$

$$1.008e^{-177.87} + 0.0692$$

$$I_2 = \frac{-12.9S - 72}{49.2S^2 + 87515} \rightarrow I_2 = \frac{-0.26S - 1.463}{S^2 + 177.87S} \rightarrow \frac{-0.26S - 1.463}{S(S + 177.87)}$$

$$f(s) = \underbrace{\frac{-0.26S}{S(S+177.87)}}_{\text{Part 1}} - \underbrace{\frac{1.463}{S(S+177.87)}}_{\text{Part 2}}$$

Part 1

$$\mathcal{L}^{-1}\left(\frac{-0.26S}{S(S+177.87)}\right)$$

$$\left(\frac{-0.26}{S+177.87}\right)$$

$$-0.26\left(\frac{1}{S+177.87}\right)$$

$$-0.26\,\mathcal{L}^{-1}\left(\frac{1}{S+177.87}\right) \rightarrow f(t) = e^{-at} = \mathcal{L}\{f(t)\} = \frac{1}{S+a}$$

$$\underline{\underline{-0.26e^{-177.8t}}}$$

Part 2

$$\mathcal{L}^{-1}\left(\frac{1.463}{S(S+177.87)}\right) \rightarrow \frac{A}{S} + \frac{B}{S+177.87}$$

Solve for A

$$F(S) = \frac{B}{S+177.87}\bigg|_{S\rightarrow 0} = \frac{1.463}{0+177.87} \rightarrow 0.0082$$

Solve for B

$$F(S) = \frac{A}{S}\bigg|_{S\rightarrow -177.87} = \frac{1.463}{-177.87} \rightarrow -0.0082$$

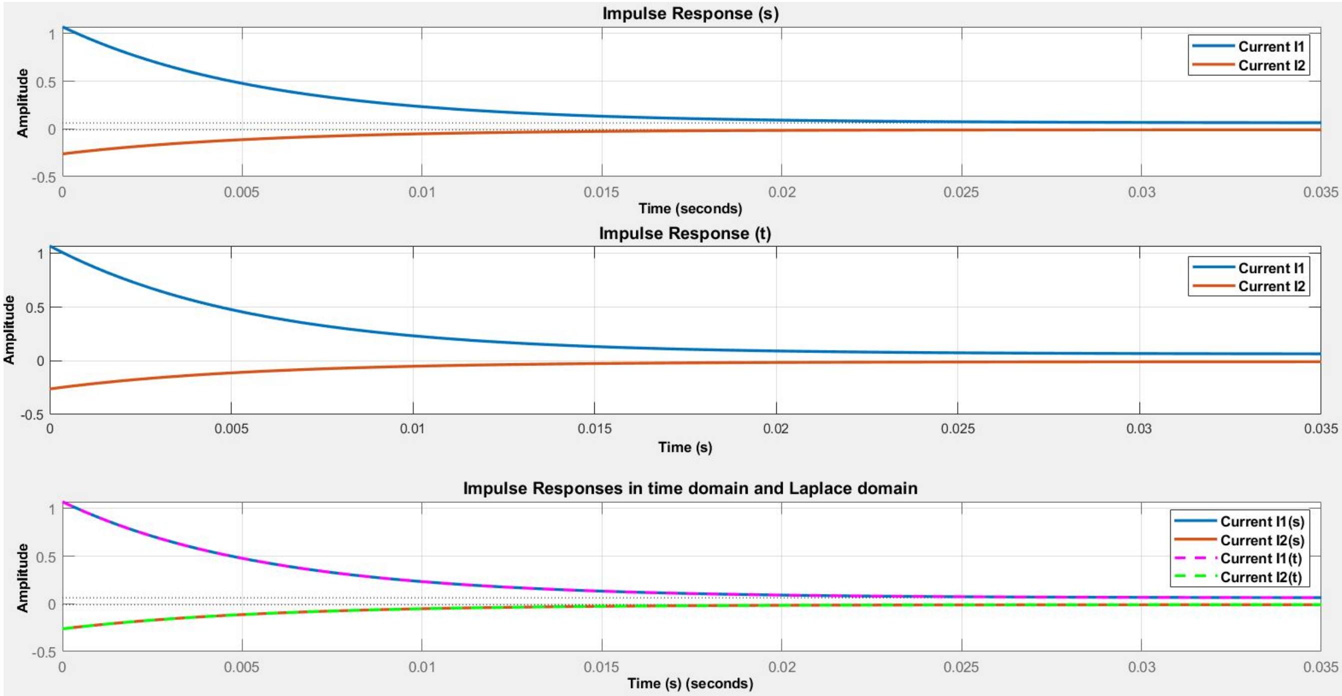
$$\frac{A}{S} \rightarrow \frac{0.0082}{S} \rightarrow \frac{a}{S} \text{ so } f(t) = 0.0082(t)$$

$$\frac{B}{S+a} \rightarrow \frac{-0.0082}{S+177.87} \rightarrow e^{-at} \rightarrow f(t) = -0.0082e^{-177.87(t)}$$

Solution Part 1 and Part 2

$$-0.26e^{-177.87t} - 0.0082t + 0.0082e^{-177.87t}$$

$$-0.252e^{-177.87t} - 0.0082t$$



Impulse Response (s)
Amplitude
Time (seconds)
Current I1
Current I2
Impulse Response (t)
Amplitude
Time (s)
Current I1
Current I2
Impulse Responses in time domain and Laplace domain
Amplitude
Time (s) (seconds)
Current I1(s)
Current I2(s)
Current I1(t)
Current I2(t)