Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.2303.12

2023, 42(3) 110-118

Simulation and analysis of ideal low pass filter response using MATLAB

Muhammad Zakir Shaikh [a] , Khakoo Mal [b, c, *], Dileep Kumar [a], Ehsan Ali [d], Bhawani Shankar Chowdhry [a], Enrique Nava Baro [e]

[a] *NCRA-Condition Monitoring Systems Lab, Mehran University of Engineering and Technology, Jamshoro, Sindh Pakistan*

[b] *Sukkur IBA University Sukkur, 65200, Sindh Pakistan*

[c] *Department of Electronic Engineering Mehran University of Engineering and Technology Jamshoro, Sindh Pakistan*

[d] *Department of Electronic Engineering Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah, Sindh Pakistan*

[e] *Universidad de Malaga, Spain*

[*] Corresponding author: Khakoo Mal, Email: khakoo.mal@iba-suk.edu.pk

Received: 07 March 2023, Accepted: 26 June 2023, Published: 01 July 2023

KEYWORDS

Filters

Communication Systems

MATLAB

Si Function

Sinc Function

ABSTRACT

Filters are the crucial element of many modern electronic systems used in control engineering, wireless communication systems, and embedded systems designs. Filters in one form or another, are considered an important subsystem in developing communication systems. The ideal filter cannot be designed through any electronic circuit and the only way is to realize using simulations tools like MATLAB. In this paper, the simulation and analysis of the ideal low pass filter is presented. The MATLAB software is used to generate and perform analysis of the step response of ideal low-pass filter (Si function) and impulse response of ideal low-pass filter (Sinc function) in time and frequency domains. The aim of the work is to provide engineering students an easier understanding of the filter design process in an appropriate way with the help of simple programming. Where, it is easy to generate signals and analyze the ideal low pass filter response. To verify achievement of learning objectives, a survey is conducted from 45 engineering students of electronic and telecommunication which elicited students' observations and interest in the learning process. Overall about 75% of students gave positive feedback, which indicates that such type of experimentation had a significant impact on the student academic outcomes.

1. Introduction

Filters play important role in designing and achieving robust performance of modern day communication systems such as radars, satellites and other navigation systems. Filters are used to separate unwanted noise and avoid interference from desired signal and provide a suitable signal. Filters are used in power supplies in order to attenuate the ripples. Similarly, in digital system digital filters are realized to minimize the noise which occurs due to coding and transmission on a noisy channel in audio/video streaming [1,2]. Filters are broadly categorized into four groups [3]: low-pass filters (LPF), high-pass filter (HPF), band-pass filters (BPF) and band-stop filters (BSF). Each of these filters used for different applications.

Engineering students of the disciplines including Electronic and Telecommunication engineering are often taught subjects like Communication systems, Signals Processing, Embedded System Design, and Control Engineering. These subjects include topics related to filters which allow them learn about how filtering process is used to remove noise from signals as it is an important function in signal conditioning. They learn about various types of filters, applications of filters, tools to design/simulate filters for required specifications [4-6]. In theory they also learn about mathematical models based on governing equations involved to design a specific filter. Generally, designer engineers perform simulations to have a quick overview of the system performance and engineering students must be familiar with problems solving methods and design flow of the systems. Various programming simulation tools such as the MATLAB allows students to learn intuitively and effectively about engineering courses through an interactive way to grip the engineering knowledge with psychomotor skills [7-9]. The MATLAB allows simulation and programming to implement and analyze engineering designs and ultimately leads to solution development [10]. Thus, considering benefits of simulation and programming in engineering education, this paper presents low-pass filter design using the MATLAB. Moreover, it presents the analysis of impulse and step responses of ideal low-pass filter for the undergraduate students of Electronic Engineering to hands-on of ideal filters' responses. The simulation would help and allow students to programmatically learn about basic concept of filters and their analysis with different standard inputs.

An ideal low-pass filter passes all signal components having frequencies less than *B* Hz with no distortion and completely attenuates signal components having frequencies greater than *B* Hz [11].

The frequency response for an ideal low-pass filter with some delay τ is:

$$H_L(f) = K\prod(f/2B)e^{-j2\pi f\tau} \quad (1)$$

The filter cutoff and bandwidth are *B*.

The impulse response of the ideal low pass-filter with delay τ is:

$$h_L(t) = F^{-1}[K\prod(f/2B)] = 2BkSinc(2Bt-\tau) \quad (2)$$

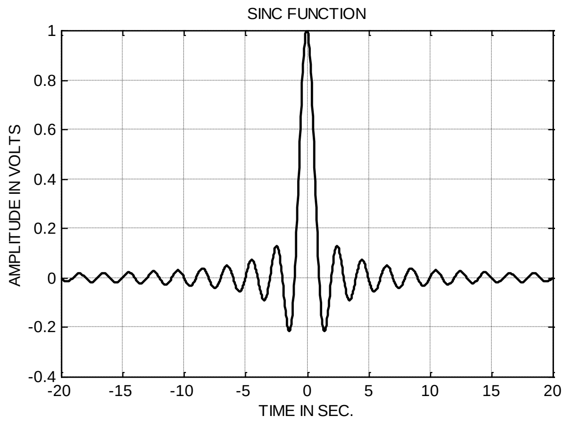
If delay time is zero, then the impulse response will be:

$$h_L(t) = 2BkSinc(2Bt) \quad (3)$$

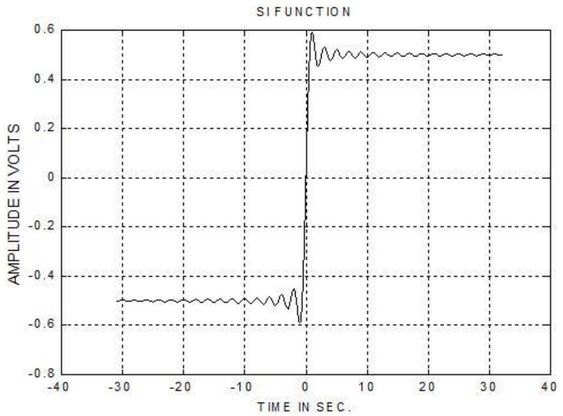
The step response of ideal low-pass filter can be expressed in terms of a well-known transcendental function, the sine integral:

$$\mathrm{Si}(z) = \frac{\omega c}{\pi}\int_0^z \frac{\sin u}{u}\,du \quad (4)$$

The time domain impulse response (Sinc function), and step response (Si function) of ideal low-pass filter are shown in Fig. 1.



(a) Time domain Impulse response



(b)Time domain step response

Fig. 1. (a) Time domain impulse response, and (b) step response of Ideal low-pass filter

Remainder of this paper is organized as follows: Section 2 provides the detailed description of the experiments sequence and student background, Section 3 lists learning outcomes and student survey, and Section 4 includes summary of experiments.

2. Experiments Sequence and Student Background

This paper is prepared for the students of Electronic, Telecommunication, and Biomedical Engineering who are essentially required to understand the working principle of a filter. The paper based on MATLAB programming that would help them to learn about low-pass filter and its response to various standard input such as impulse signal. Firstly, both the impulse response and step response of ideal low-pass filter are simulated through the MATLAB, then Fast Fourier Transform (FFT) is applied to observe the frequency response.

Practical-I: Simulation of Ideal Low Pass Filter and Its Impluse Response

As the impulse response of ideal low-pass filter (Sinc function) contributes important role in scientific experiments like in image reconstruction, nuclear magnetic resonance NMR spectroscopy and condition monitoring systems [12,13]. The impulse response of ideal low-pass filter or Sinc function will be simulated in MATLAB by built-in Sinc function. The programming code is given in Appendix 1.

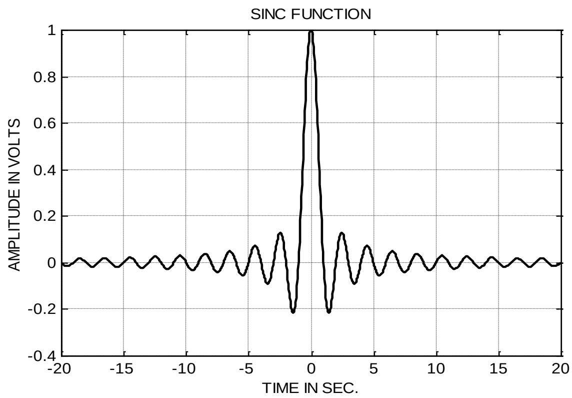
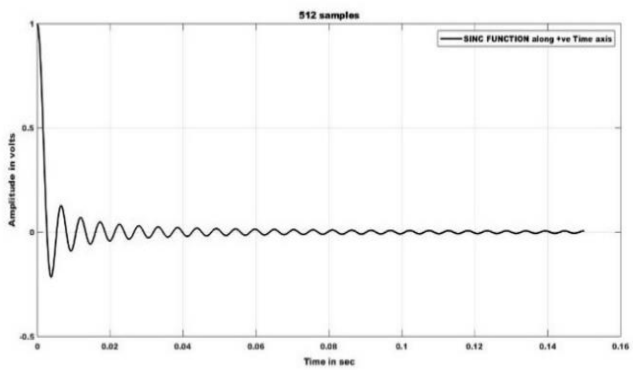
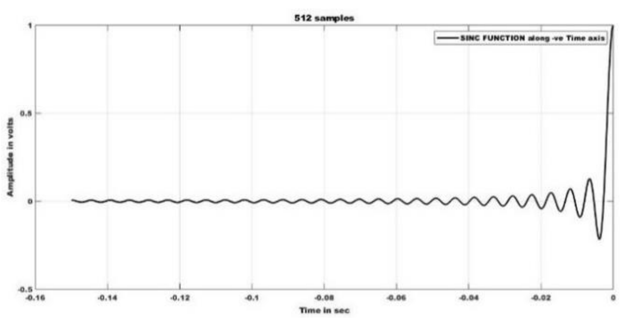


Fig. 2. Mathematically generated Sinc function

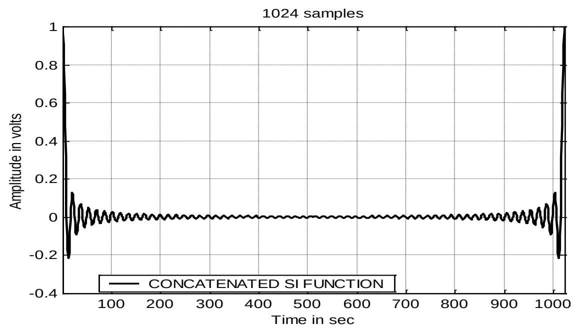
The above Sinc function is mathematically generated but is non-causal function, therefore now it will be cyclically rotated for restoring the time origin. The function with re-established time origin will not introduce any time shift in the function. In MATLAB the time-origin will be re-established by cat (concatenation) function. The programming code is given in Appendix 2.



(a) Positive time half Sinc function



(b) Negative time half Sinc function



(c) Concatenated Sinc function

Fig. 3. Concatenation (cyclic rotation) of Sinc Function

The above concatenated Sinc function is not tapered, therefore it contains noise signal, which in turn will cause incorrect frequency response. Therefore, we first taper Sinc function and then get Fourier Transform. In MATLAB we taper Sinc function by the code given in Appendix 3.

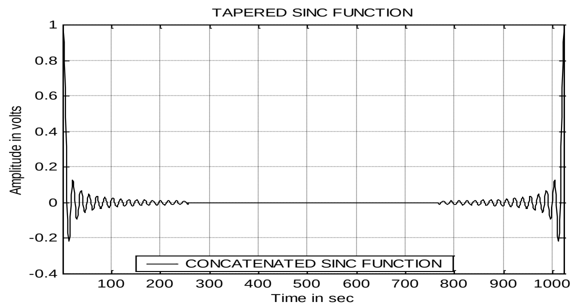


Fig. 4. Tapered and Concatenated Sinc function

Practical-II: Simulation of Step Response of Ideal Low-Pass Filter

This practical covers step response analysis of ideal low-pass filter (Si function) which can be simulated by two ways in the MATLAB.

(a) Step response by taking Integral of built-in Sinc function

The Si function can be generated by taking integration of Sinc function, because the Si function is not built-in. The integration will be taken by 'Integral'. The program code is given in Appendix 4.

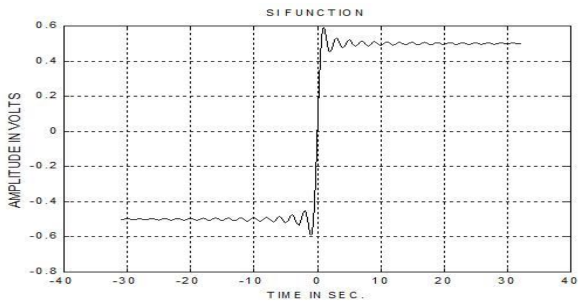


Fig. 5. Si function generated by integrating Sinc function

(b) Step response by combining power series and asymptotic series:

The Si function is not available in closed form, therefore it can be represented in series form [14], for small values of the argument by a power series:

Si (x) =x-x^3/3.3!+x^5/5.5!- x^7/7.7!+……. (5)

And for large values by an asymptotic series:

Si(x) = π /2-[1!/x2-3!/x4+5!/x6 - . . .]sin(x)-[0!/x-2!/x3+4!/x5 ...]cos(x) (6)

Where the asymptotic series is semi-convergent, which means that as more terms in the series are taken, initially it converges, but after the certain terms it diverges. The term at which the series starts to diverge depends on the value of x for which the series is being evaluated. Therefore, the series should be used only up to the point where the series stops converging, and the accuracy improves as the value of x increases. This slowness of convergence is illustrated in the asymptotic series, where it can be seen that for large arguments the envelope of oscillations varies as 1/x.

We have simulated these series in MATLAB by the programming code given in Appendix 5.

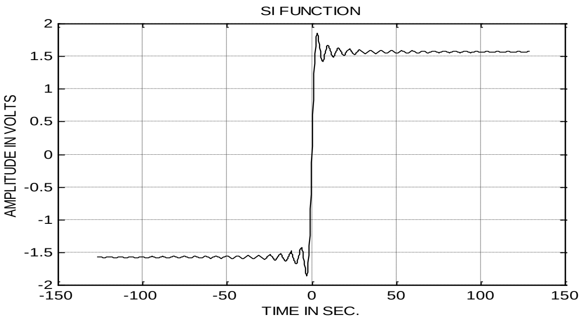


Fig. 6. Si function generated by power series and asymptotic series

Practical-III: Simulation of the Effects of Time Delay

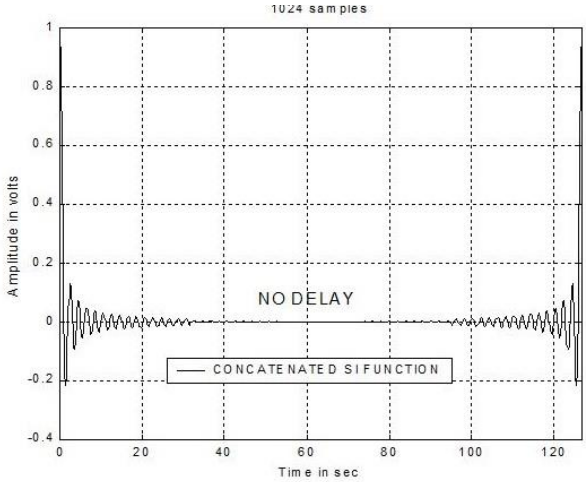
Now we analysis the effects of time delay in Sinc function (impulse response of ideal low-pass filter). In MATLB we have introduced time delays of one sample, two sample, half sample and quarter sample. We took Fourier transform of delayed Sinc function and the Fourier transformed imaginary part of delayed Sinc function wave form exhibited a clear peak with significant imaginary error. The time advance also introduced in Sinc function and taken its Fourier transforms.

Time delay leaves the modulus of the Fourier transform unchanged but introduces a phase shift in its Fourier transform, which is a linear function of time delay [15-17]. The real and imaginary parts of Fourier transformed Sinc function are given below.

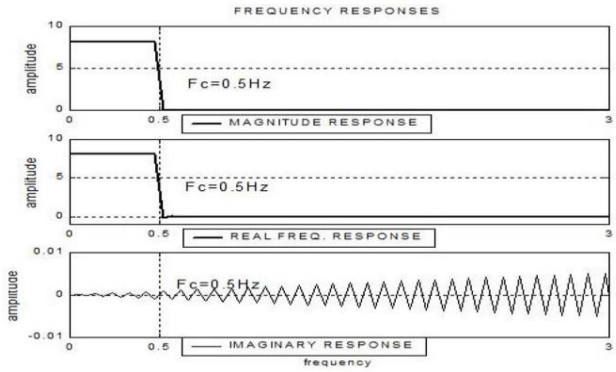
$$\mathrm{Re}(\omega)= H(\omega)\cos(\omega\tau) \qquad (7)$$

$$\mathrm{Im}(\omega)= -H(\omega)\sin(\omega\tau) \qquad (8)$$

Therefore, by inserting the values of H(ω), ω and τ, peaks in real and imaginary parts can be calculated. The simulation of effects of delay is given in the MATLAB programing codes given in Appendix 6 and effect is shown in figures.



(a) Time domain Sinc function



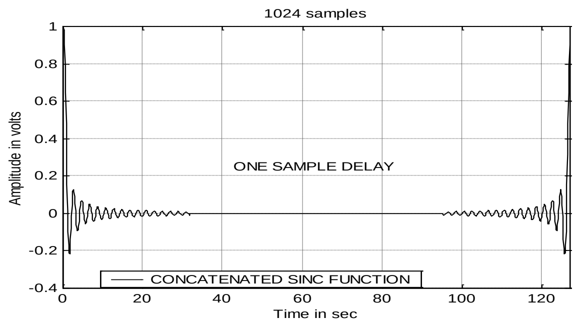
(b) Frequency domain Sinc function

Fig. 7. Time domain and frequency domain Sinc function with no delay

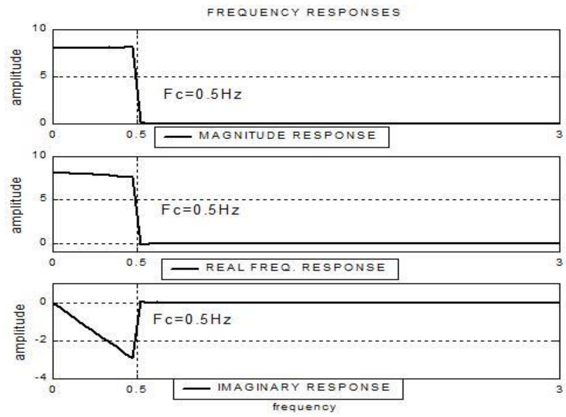
Here now we have shown only waveforms and skipped programing codes because these are similar to the program given in Appendix 6, only delay is introduced.

One Sample Delay

% Sampling time (Ts) =0.1233



(a) Time domain Sinc function with one sample delay

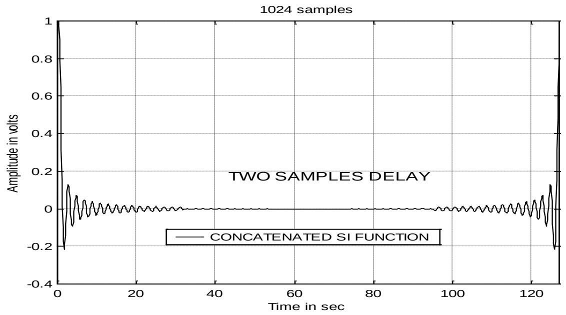


(b) Frequency domain Sinc function with one sample delay

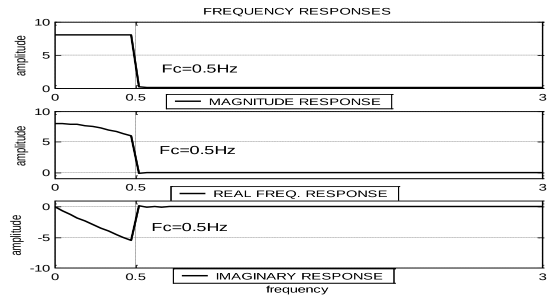
Fig. 8. Time domain and frequency domain Sinc function with one sample delay

Two Samples Delay

% Two Samples Delay (2Ts) = 0.2466.



(a) Time domain Sinc function with two samples delay

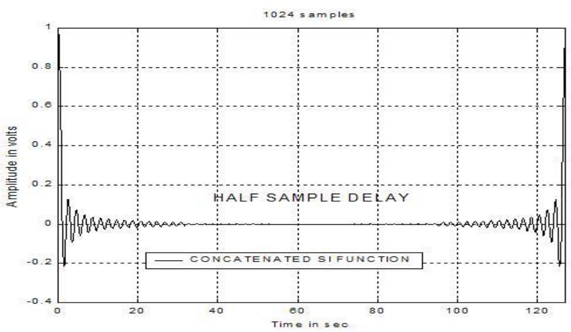


(b) Frequency domain Sinc function with two samples delay

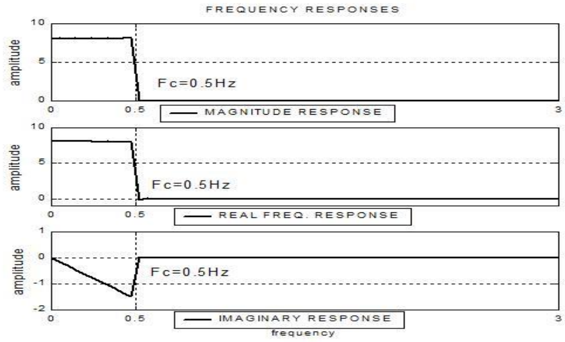
Fig. 9. Time domain and frequency domain Sinc function with two samples delay

Half Sample Delay

% Half-A- Sample Delay (1/2Ts) = 0.06165.



(a) Time domain Sinc function with half sample delay

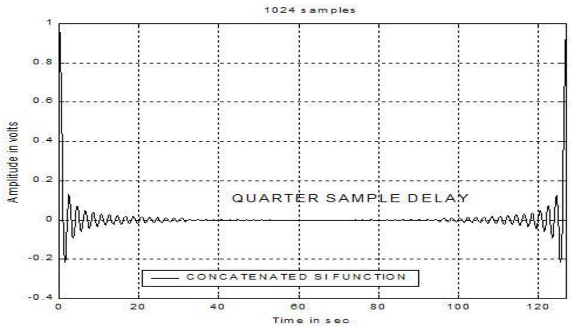


(b) Frequency domain Sinc function with half sample delay

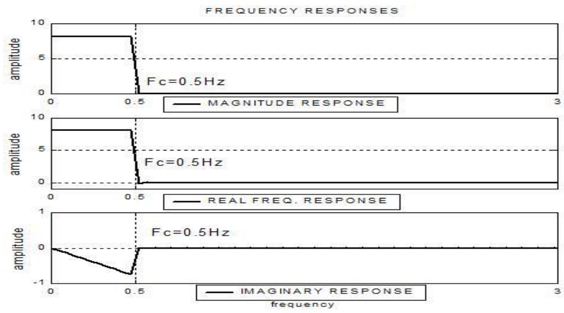
Fig. 10. Time domain and frequency domain Sinc function with half sample delay

Quarter Sample Delay

% Quarter Sample Delay (1/4 Ts) = 0.030825.



(a) Time domain Sinc function with quarter sample delay

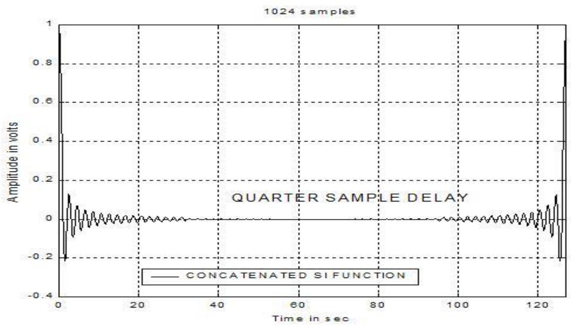


(b) Frequency domain Sinc function with quarter sample delay

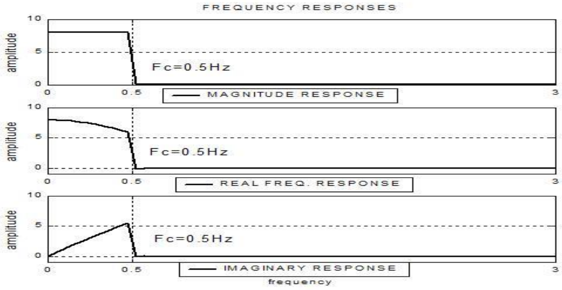
Fig. 11. Time domain and frequency domain Sinc function with quarter sample delay

Two Sample Advance

% Two Samples Advance (2Ts) = 0.2466.



(a) Time domain Sinc function with two samples advance



(b) Frequency domain Sinc function with two sample advance

Fig. 12. Time domain and frequency domain Sinc function with two samples advance

The delay effect on imaginary error is clear in above figures (Fig. 7, 8, 9, 10, 11), the advance time has same effect on frequency response only sign is opposite as shown in Fig. 12. By putting the values of $\omega=2\pi F_c$, where $F_c=0.5$ and $\tau=0.030825$ (quarter sample), 0.06165 (half sample), 0.1233 (one sample) and 0.2466 (two sample) in equation (8) the imaginary error is calculated as given in Table 1 and the magnitude of imaginary error against delay is plotted in Fig. 13.

Table 1

Imaginary error

S. No.	Delay (samples)	|Imaginary Error|
1.	0	0
2.	0.25	0.7
3.	0.5	1.4
4.	1	2.8
5.	2	5.6

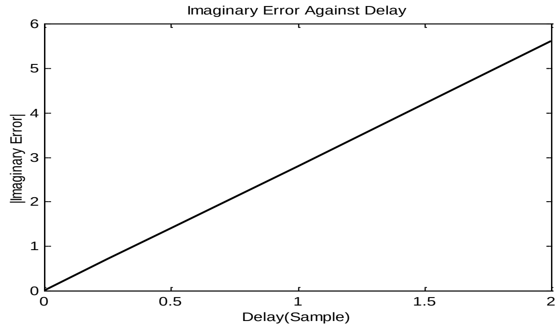


Fig. 13. Magnitude of imaginary error against delay

3. Learning Outcomes and Student Survey

This paper covering three practical will enable students to achieve following outcomes:

1. Students are able to understand the basics of filters.
2. This study will enable students to learn about filters and their applications. It will give them a simulation based exposure of filters.
3. The simulation based demonstration of low pass filter will motivate students to understand and perform analysis of filters using the MATLAB.
4. It will let them to learn the MATLAB programming for signal processing.
5. The students would be able to perform filter design in software which will help them to develop physical filter circuit based on their observations in simulation based practical.

Moreover, a survey based on learning outcomes was conducted from 45 students of the Electronic Engineering program and the achieved results are provided in the Table 2.

Table 2

Students survey of the practical

S.No	Query	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1.	Understanding of what are the filters and their purpose.	1%	9%	16%	33%	41%
2.	Developing a filter after performing this practical.	1.5%	8%	13.5%	30%	47%
3.	The provided information is enough for understanding the basic concept of filter.	0.5%	12.5%	9%	41%	37%
4.	The instructor assisted while learning about the filters and implementing the code.	2.5%	9%	8.5%	38%	42%
5.	Ability to understand all the parameters of a filter and their role in implementation of a filter.	1%	8%	11.5%	40%	39.5%

The ideal low-pass has impulse response is Sinc function and step response is Si function, time delay leaves the modulus of the Fourier transform unchanged but introduces a phase shift in its Fourier transform, which is a linear function of time delay. The step response of ideal low-pass filter (Si function) is simulated in MATLAB by two ways (i) taking integration of built-in Sinc function and (ii) combining power series and asymptotic series. Mathematically generated Sinc function in MATLAB is non-causal function, therefore we have cyclically rotated Sinc function by cat (catenation) function for restoring time origin and then tapered. Finally delay effect in Sinc function was analyzed by introducing different delays, also we have analyzed the time advance in Sinc function.

4. Summary

This paper presented an approach related to simulation and analysis of an ideal low-pass filter which is a key component of various communication systems and embedded systems. The response analysis of the filter was performed with different input signals such as impulse and step signals and frequency domain analysis of the achieved output. The effect of time delay in impulse response is also analyzed and observed that it introduces phase shift in its frequency response. The MATLAB based experiments were conducted to teach students about the basics of filters and their response with different signals. Moreover, students were able to learn about the filter through programmatically varying different parameters of the filer. The experiments allowed students of Electronic Engineering to achieve different learning outcomes and it was confirmed through a survey conducted by asking five core outcomes from the students. The survey results demonstrated that more than 75% students were satisfied the contents, experimentation, and the approach used to teach them about filers.

Funding

Not applicable.

Data availability

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

5. References

[1] D. Báez-López, D. Báez-Villegas, R. Alcántara, J.J. Romero, T. Escalante, “Package for filter design based on MATLAB”, Comp App in Eng Edu 9 (4):259-264, 2001.

[2] D. Báez-López, G. Cabrera, P. Martínez, G. Reyes, R. Jaroszynski, J. Luna, J. Ramírez, G. E. Flores-Verdad, “Multimedia-based analog and digital filter design”, Comp App in Eng Edu. 6 (1):1-8, 1998.

[3] S. Mandal, “Some important simulation software tools for a student of electronics engineering”, Glob Jour on Adv in Eng and Sci. 3 (1), 2017.

[4] L. Stanković, E. Sejdić, S. Stanković, et al., “A tutorial on sparse signal reconstruction and its applications in signal processing”, Circuits Syst Signal Process 38, 1206–1263, 2019. https://doi.org/10.1007/s00034-018-0909-2

[5] R. Krneta, M. Bjekic, A. Dostanic, M. Acovic, “Influence of use of the MATLAB as underlying technology on development of the curriculum for the course in signals and systems”, 14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services, United States, IEEE, pp 165-168

[6] A. Atamas, I. Slipukhina, I. Chernetskyi, Y. Shykhovtsev, “Supercapacitor as a didactic means of teaching students fundamentals of electronics in the NI multisim environment”, Physical and Math Edu. 2(24). P. 7-12, 2020.

[7] L. Aksoy, P. Flores, and J. Monteiro, “Tutorial on multiplierless design of FIR filters: algorithms and architectures”, Circuits Syst Signal Process 33, 1689–1719, 2014. https://doi.org/10.1007/s00034-013-9727-8

[8] L. Sevgi, “Modeling and simulation concepts in engineering education: virtual tools”, Turkish Journal of Elect Eng and Comp Sci 14 (1):113-127, 2006.

[9] S. Lee, J. Kim, W. Lee, “Analysis of factors affecting achievement in maker programming education in the age of wireless communication”, Wire Pers Comm 93 (1):187-209, 2017.

[10] H. G. Wright, T. B. Welch, and M. G. Morrow, “Leveraging student knowledge of DSP for optical engineering”, Proceedings of IEEE Signal Processing and Signal Processing Education Workshop, Salt Lake City, Utah, USA, pp. 148–153, 2015.

[11] A. de Cheveigné, I. Nelken, “Filters: when, why, and how (not) to use them”, Neuron 102 (2):280-293, 2019.

[12] K. Mal, I. Hussain, B. S. Chowdhry, T. D. Memon, “Extended kalman filter for estimation of contact forces at wheel-rail interface”, 3C Technologia (Special Issue): 279-301, 2020. doi:10.17993/3ctecno.2020.specialissue5.279-301

[13] D. D. Traficante, M. M. McGregor, “Simple laboratory experiments to observe the manifestations of the sinc function”, Concepts

in Magnetic Resonance: An Educational Journal 14 (5):308-325, 2002.

[14] W. Flügge W, “Tables of transcendental functions”, Pergamon Press, 1954.

[15] B. Chowdhry, S. Shahi, J. Brignell, “Application of unorthodox stimuli in computer measurement: dielectric relaxation as an example”, Journal of Physics E: Sci Instr. 21 (3):259, 1988.

[16] K. Mal, T. Memon, I. Hussain, B. Chowdhry, “FPGA implementation of extended kalman filter for parameters estimation of railway wheelset”, CMC, 74(2), pp. 3351–3370, 2022.

[17] D. R. Lorenza, R. Bonelli, A. Gerosa, “Formative tutoring: a program for the empowerment of engineering students”, IEEE Transactions on Education , 66(2), 2022.

APPENDIX 1: PROGRAM 1 of *Non-Causal Sinc Function*

```
t=linspace(-20,20,1024);
y=sinc(t);
plot(t,y); grid
title('SINC FUNCTION');
xlabel('TIME IN SEC.');
ylabel('AMPLITUDE IN VOLTS');
```

APPENDIX 2: PROGRAM 2 of *Concatenated Sinc Function*

```
t=linspace(0,0.15,512); f1=60; y=Sinc(2*pi*f1.*t);
figure(1); plot(t,y);  grid
xlabel('Time in sec'); ylabel('Amplitude in volts');
axis([0,0.16,-0.5,1]);  title('512 samples')
legend('SINC FUNCTION along +ve Time axis')
figure(2); t1=linspace(-0.15,0,512);
y1=Sinc(2*pi*f1.*t1);
plot(t1,y1);  grid
xlabel('Time in sec');  ylabel('Amplitude in volts');
axis([-0.16, 0, -0.5,1]);  title('512 samples')
legend('SINC FUNCTION along -ve Time axis')
figure(3); y2=cat(2, y, y1); plot(y2(:));  grid
xlabel('Time in sec'); ylabel('Amplitude in volts');
title(‘1024 samples’);
legend('CONCATENATED SINC FUNCTION')
```

APPENDIX 3: PROGRAM 3 of *Tapered Sinc Function*

```
t=linspace(0,63,512); y=Sinc(t);
for u=1:length(y);
if u > 256,  y(u)=y(u)*0.05;
else y(u)=y(u);
end
end
t1=linspace(-63,0,512);
y1=sinc(t1);
for u1=1:length(y1);
if u1 <256,  y1(u1)=y1(u1)*0.05;
else y1(u1)=y1(u1);
end
end
y2=cat(2,y,y1);
plot(y2(:)); grid
xlabel('Time in sec'); ylabel('Amplitude in volts')
title('1024 samples'); axis([1,1024,-0.4,1])
legend('CONCATENATED SINC FUNCTION ')
```

APPENDIX 4: PROGRAM 4

```
t=linspace(-31,32,1024);
for u=1:length(t),
y1=@(t) Sinc(t);
y(u)=integral(y1,0,t(u))
end
plot(t,y);  grid
title('SI FUNCTION');
xlabel('TIME IN SEC.');  ylabel('AMPLITUDE IN VOLTS');
```

APPENDIX 5: PROGRAM 5

```
x=linspace(-127,128,1024); %initiating 1024 samples .
for u=1:length(x);
if x(u)<12 & x(u)>-12,       % For power series----------.
```

```
for n=1:1.0:15; % Calculate 1st 15 terms--.
a=(-1)^n; b=(-1)*a;  c=(2*n)-1; d=(x(u))^c;
e=c*(factorial(c));
f=b*d;  g(n)=f/e;
end
si(u)=sum(g);
elseif  x(u)>=12,  %  Asymptotic  series  0f  +ve argument.
forn=1:1.0:15;  a=(-1)^n; b=(-1)*a;
c=(2*n); c1=c-1; d=(x(u))^c;  e=(factorial(c1));
f=b*e;  g(n)=f/d;  g1=sum(g); h=g1*sin(x(u));
i=2*(n-1); j=factorial(i);  k=(x(u))^c1; l=b*j;
m(n)=l/k; m1=sum(m); o=m1*cos(x(u));
end
si(u)=pi/2-h-o;
else %For asymptotic series of -ve argument.
for n=1:1.0:15;
a=(-1)^n; b=(-1)*a;
c=(2*n); c1=c-1; d=(x(u))^c;e=(factorial(c1));
f=b*e; g(n)=f/d;
g1=sum(g); h=g1*sin(x(u));
i=2*(n-1); j=factorial(i);  k=(x(u))^c1;
l=b*j; m(n)=l/k;  m1=sum(m); o=m1*cos(x(u));
end
si(u)=-(pi/2-(h-o));
end
end
plot(x,si), grid
title('SI FUNCTION');
xlabel('TIME  IN  SEC.');  ylabel('AMPLITUDE  IN VOLTS');
```

APPENDIX 6: PROGRAM 6 of *Without Delay*

```
t=linspace(0,63,512);   %initiating   +VE   time 512Samples
y=sinc(t); for u=1:length(y);
if u > 256,  y(u)=y(u)*0.05;
else y(u)=y(u);
end
end
t1=linspace(-63,0,512); %initiating –ve 512 Samples
y1=sinc(t1);
for u1=1:length(y1);
if u1 <256,  y1(u1)=y1(u1)*0.05;
else y1(u1)=y1(u1);
end
end
figure(1);  y2=(cat(2,y,y1)); % For Concatenate
t2=linspace(0,127,1024);   %All   +VE   time   1024 Samples
plot(t2,y2); grid; xlabel('Time in sec')
ylabel('Amplitude in volts')
title('1024 samples')
legend('CONCATENATED SINC FUNCTION ')
axis([0,127,-.4,1])
f=linspace(0,6,128);   %initiating   freq   range 128Samples
for z=1:length(f);
for z1=1:length(y2);
a(z1)=y2(z1)*exp(-j*2*pi*f(z)*t2(z1)); a1=sum(a);
end
b(z)=a1-1;
end
figure(2); subplot(311)
plot(f,abs(b)); grid; axis([0 3 0 10])
set(gca,'XTick',[0 0.5 3]); ylabel('amplitude')
title('FREQUENCY RESPONSES')
legend('MAGNITUDE RESPONSE')
subplot(312); plot(f,real(b)); grid
axis([0 3 -1 10]); set(gca,'XTick',[0 0.5 3])
ylabel('amplitude');
legend('REAL FREQ. RESPONSE')
subplot(313); plot(f,imag(b)); grid
axis([0 3 -0.01 0.01]); set(gca,'XTick',[0 0.5 3])
xlabel('frequency'); ylabel('amplitude')
legend('IMAGINARY RESPONSE')
```