

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA ĐIỆN TỬ - VIỄN THÔNG**



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**Môn học: Digital Signal Processing**

**Giảng viên: Trần Thị Thảo Nguyên**

**Lớp: 21DTV2**

**Người thực hiện: Nguyễn Tiến Đại**

**Năm học: 2023 – 2024**

# MỤC LỤC

<b>MỤC LỤC .....</b>	<b>i</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>iii</b>
<b>DANH MỤC BẢNG BIỂU.....</b>	<b>iv</b>
<b>DANH MỤC VIẾT TẮT .....</b>	<b>v</b>
<b>CHƯƠNG 1: GIỚI THIỆU .....</b>	<b>1</b>
1. Đề án môn học.....	1
2. Mục tiêu đề án .....	1
<b>CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....</b>	<b>2</b>
1. Lấy mẫu (Sampling) .....	3
1.1 Khái niệm.....	3
1.2 Định lý lấy mẫu Nyquist-Shannon: .....	5
2. Hiện tượng Aliasing (chồng lấn phổ) .....	6
2.1 Khái niệm Aliasing .....	6
2.2 Nguyên nhân gây ra Aliasing.....	7
2.3 Ảnh hưởng của Aliasing .....	7
2.4 Ngăn ngừa Aliasing .....	7
3. Khôi phục tín hiệu (Reconstruction).....	8
3.1 Zero- order- hold (ZOH) interpolation: .....	8
3.2 First-order-hold (FOH) interpolation:.....	9
3.3 Khôi phục hình thang : .....	12
<b>CHƯƠNG 3: MÔ PHỎNG MATLAB VÀ JUPYTER NOTEBOOK .....</b>	<b>14</b>
3.1.1 Mô phỏng Matlab .....	14
3.1.2 Mô phỏng bằng Simulink .....	18

3.2 Mô phỏng Jupyter Notebook .....	19
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>27</b>



## DANH MỤC HÌNH ẢNH



## DANH MỤC BẢNG BIỂU



## DANH MỤC VIẾT TẮT



## CHƯƠNG 1: GIỚI THIỆU

### 1. Đề án môn học

Trong lĩnh vực xử lý tín hiệu số, quá trình lấy mẫu và khôi phục tín hiệu đóng vai trò then chốt để chuyển đổi và xử lý các tín hiệu liên tục trong thế giới thực thành dạng số nhị phân mà máy tính có thể xử lý. Những bước này không chỉ cần thiết để lưu trữ và truyền tín hiệu hiệu quả mà còn để đảm bảo rằng thông tin gốc được giữ trọn vẹn. (1) Lấy mẫu tín hiệu chuyển đổi tín hiệu liên tục (analog) thành tín hiệu rời rạc (digital) bằng bộ ADC. (2) Khôi phục tín hiệu thì quá trình ngược lại so với lấy mẫu → chuyển đổi tín hiệu rời rạc (digital) trở lại dạng liên tục (analog) thông qua DAC. Hai quá trình trên ứng dụng rất nhiều vào các thiết bị đời sống như: âm thanh và video số, truyền thông số, xử lý ảnh... Trong quá trình thực hiện đồ án, em xin chân thành cảm ơn cô Nguyễn đã hướng dẫn em những kiến thức cốt lõi và nhiệt tình giải đáp những vấn đề em gặp phải để trọn vẹn thực hiện đồ án này.

### 2. Mục tiêu đồ án

- ✚ Hiểu rõ lý thuyết về lấy mẫu và khôi phục tín hiệu → định lý lấy mẫu Nyquist-Shannon.
- ✚ Thiết kế và thực hiện mô phỏng quá trình lấy mẫu và khôi phục tín hiệu → Sử dụng Matlab Simulink.
- ✚ Phân tích và đánh giá kết quả mô phỏng → So sánh và đánh giá tín hiệu gốc và tín hiệu sau khi khôi phục để kiểm tra độ chính xác.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

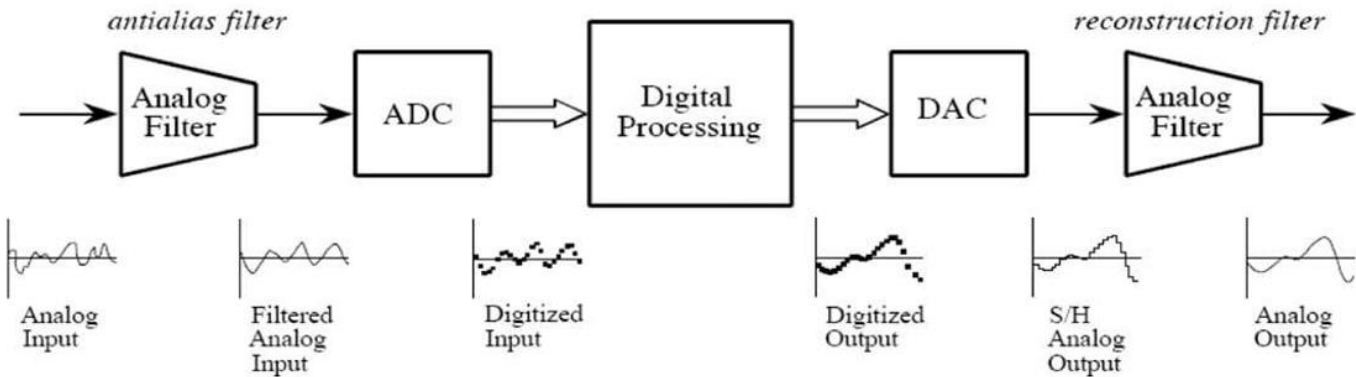


Fig 2.1 Block Diagram of Digital Signal Processing

Tín hiệu tương tự được chuyển đổi sang tín hiệu số thông qua bộ ADC qua từng bước sau:

- Sampling
- Quantising
- Coding

Bộ lấy mẫu tín hiệu đầu vào với khoảng thời gian lấy mẫu  $T$ . Tín hiệu đầu ra có biên độ rời rạc theo thời gian nhưng liên tục. Đầu ra của bộ lấy mẫu được đưa tới bộ lượng tử hoá. Nó chuyển đổi tín hiệu thành tín hiệu rời rạc theo thời gian, biên độ rời rạc. Bước cuối cùng là mã hoá sẽ giải mã từng giá trị mẫu được lượng tử hoá từ bộ lọc số.

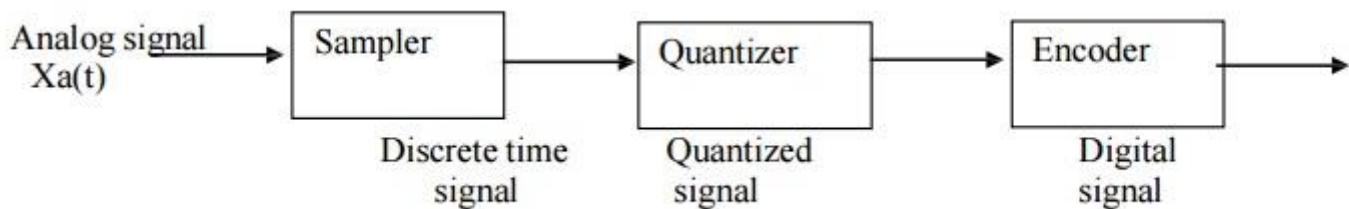


Fig 2.2 Block Diagram of Analog to Digital Conversion



1. Lấy mẫu: Đây là sự chuyển đổi tín hiệu thời gian liên tục thành tín hiệu thời gian rời rạc thu được bằng cách lấy “mẫu” tín hiệu thời gian liên tục tại các thời điểm rời rạc. Do đó, nếu  $x_a(t)$  là đầu vào của bộ lấy mẫu thì đầu ra là  $x_a(nT) = x(n)$  trong đó  $T$  gọi là thời gian lấy mẫu.
2. Lượng tử hoá: Đây là sự chuyển đổi tín hiệu có giá trị liên tục theo thời gian rời rạc thành tín hiệu có giá trị rời rạc theo thời gian. Giá trị của mỗi mẫu tín hiệu được biểu thị bằng một giá trị được chọn từ một tập hữu hạn các giá trị có thể. Sự khác biệt giữa mẫu không được lượng tử hoá  $x(n)$  và đầu ra được lượng tử hoá  $x_q(n)$  được gọi là lỗi lượng tử hoá.
3. Aliasing là hiệu ứng khiến các tín hiệu khác nhau trở nên không thể phân biệt được khi lấy mẫu. Aliasing tạo ra tần số sai cùng với tần số chính xác khi thực hiện lấy mẫu tần số.

## 1. Lấy mẫu (Sampling)

### 1.1 Khái niệm

Lấy mẫu là quá trình chuyển đổi 1 tín hiệu liên tục theo thời gian (analog) thành tín hiệu rời rạc (digital) bằng cách lấy các giá trị của tín hiệu tạo các khoảng thời gian đều đặn.

$$X(n) = X_a(nT) \quad -\infty < n < \infty$$

Trong đó,  $X(n)$  là tín hiệu theo thời gian rời rạc thu được bằng cách lấy mẫu của tín hiệu tương tự  $X_a(t)$  cứ sau  $T$  giây. Khoảng thời gian  $T$  giữa các mẫu liên tục được gọi là chu kì lấy mẫu hoặc khoảng mẫu và nghịch đảo của nó là  $\frac{1}{T} = F_s$  gọi là tốc độ lấy mẫu

$$T = nT = \frac{n}{F_s}$$

Xét một tín hiệu hình sin tương tự có dạng:  $X_a(t) = A \cdot \cos(2\pi Ft + \varphi)$

Khi chu kì lấy mẫu tại  $F_s = \frac{1}{T}$  mỗi giây thì  $X_a(nT) = x(n) = A \cos(2\pi Fnt + \varphi) = A \cos(2\pi n \frac{F}{F_s} + \varphi)$  bằng cách so sánh cả 2 phương trình ta thu được mối quan hệ giữa các biến tần số

$$f = F/F_s \text{ or } \omega = \Omega T$$

trong đó:  $F_s \rightarrow$  Tần số lấy mẫu

$F \rightarrow$  tần số tín hiệu analog

$f \rightarrow$  tần số tín hiệu số

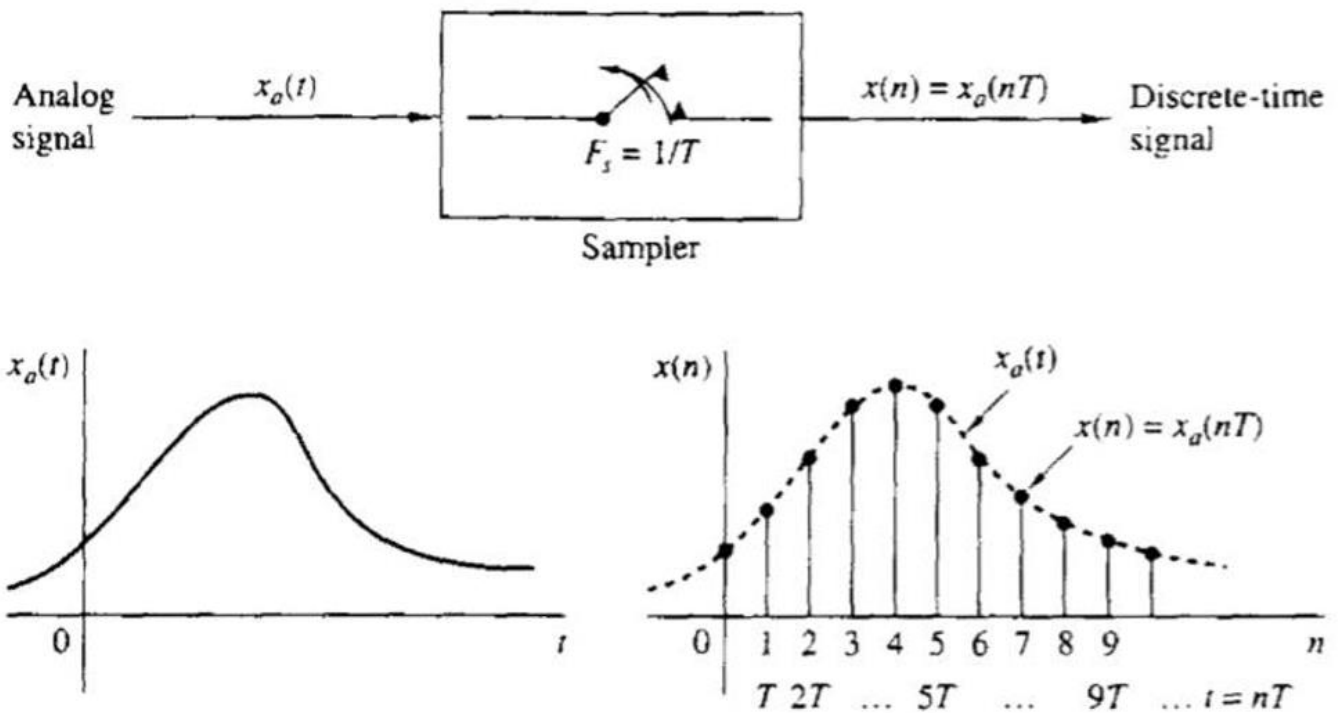


Fig 2.3 Sampling Model

Dải giá trị tần số đối với tín hiệu sin liên tục

$$-\infty < F < \infty$$

$$-\infty < \Omega < \infty$$

Đối với tín hiệu hình sin rời rạc

$$-1/2 < f < 1/2$$

$$-\pi < \omega < \pi$$

Dải tần số của sóng sin thời gian liên tục khi lấy mẫu ở tốc độ  $F_s = 1/T$

$$-1/2T = -F_s/2 < F < F_s/2 = 1/2T$$

$$-\pi/T = -\pi F_s \leq \Omega \leq \pi F_s = \pi/T$$

## 1.2 Định lý lấy mẫu Nyquist-Shannon:

### a. Định lý

Định lý lấy mẫu đảm bảo rằng tín hiệu tương tự có thể được phục hồi hoàn toàn miễn là tốc độ lấy mẫu ít nhất lớn gấp đôi thành phần tần số cao nhất của tín hiệu tương tự được lấy mẫu.

$$F_s \geq 2F_{\max}$$

$F_s \rightarrow$  tần số lấy mẫu

$F_{\max} \rightarrow$  tần số cao nhất của tín hiệu

### b. Tần số lấy mẫu

Tần số lấy mẫu ( $F_s$ ) chính là số mẫu được lấy trong 1 giây.

Vd: Đối với tín hiệu âm thanh với tần số tối đa 20kHz, tần số lấy mẫu phải ít nhất 40kHz để đảm bảo tín hiệu có thể khôi phục chính xác.

### c. Ví dụ

1. For the following analog signal, find the Nyquist sampling rate, also determine the digital signal frequency and the digital signal  $x(t)=3\cos(70\pi t)$

The maximum frequency component is  $x(t)$  is

$$F_{\max} = \frac{70\pi}{2\pi} = 35\text{Hz}$$

Therefore according to Nyquist, we need a sampling rate of

$$F_s = 2F_{\max} = 70\text{Hz}$$

The digital signal would have a frequency

$$W = 2\pi \frac{25}{70} = \pi$$

The digital signal can be represented as

$$x[n] = 3\cos(\pi n)$$

2. Find the sampling frequency of the following signal.

$$s(t) = 3\cos(50\pi t) + 10\sin(300\pi t) - \cos(100\pi t)$$

$$F1 = \frac{50\pi}{2\pi} = 25\text{Hz}$$

$$F2 = \frac{300\pi}{2\pi} = 150\text{Hz}$$

$$F3 = \frac{100\pi}{2\pi} = 50\text{Hz}$$

So sampling frequency should be  $f_s > 300\text{Hz}$

## 2. Hiện tượng Aliasing (chồng lấn phổ)

### 2.1 Khái niệm Aliasing

Aliasing xảy ra khi tần số lấy mẫu của một tín hiệu không đủ cao để bắt đầy đủ chi tiết của tín hiệu gốc, dẫn đến tín hiệu số hoá bị méo mó hoặc chứa thành phần tần số sai lệch so với tín hiệu gốc.

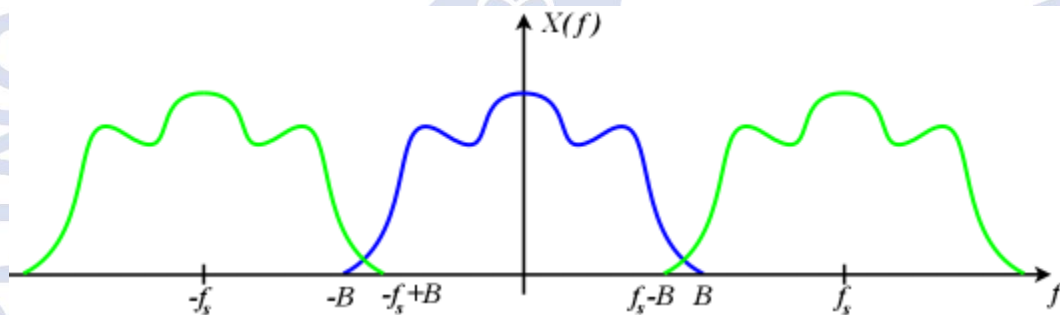


Fig 2.4 The blue sampled signal is sufficiently bandlimited

- Đối với tần số cao nhất B cho trước, chúng ta có giới hạn dưới của tần số lấy mẫu:  $2B$  hoặc tốc độ Nyquist. Ví dụ: đối với tín hiệu có tần số tối đa là 16 KHz, chúng ta cần tốc độ lấy mẫu là 32 KHz.
- Đối với một tỷ lệ lấy mẫu nhất định, chúng ta có giới hạn trên cho các thành phần tần số:  $B < f_s/2$ , hoặc tần số Nyquist hoặc  $F_{\max}$ . Ví dụ: đối với một tín hiệu có tỷ lệ lấy mẫu là 48 KHz, chúng ta có thể lấy mẫu tín hiệu lên đến 24 KHz.

- Trên thực tế, tín hiệu không bao giờ có thể được giới hạn băng tần hoàn hảo. Ngay cả khi có thể thực hiện khôi phục lý tưởng, tín hiệu được khôi phục sẽ không chính xác là tín hiệu gốc. Lỗi tương ứng với lỗi giới hạn băng tần được gọi là răng cưa.

## 2.2 Nguyên nhân gây ra Aliasing

- Tần số lấy mẫu không đủ
- Chưa sử dụng bộ lọc chống aliasing (anti-aliasing filter)

## 2.3 Ảnh hưởng của Aliasing

- Tín hiệu méo mó
- Thông tin bị mất

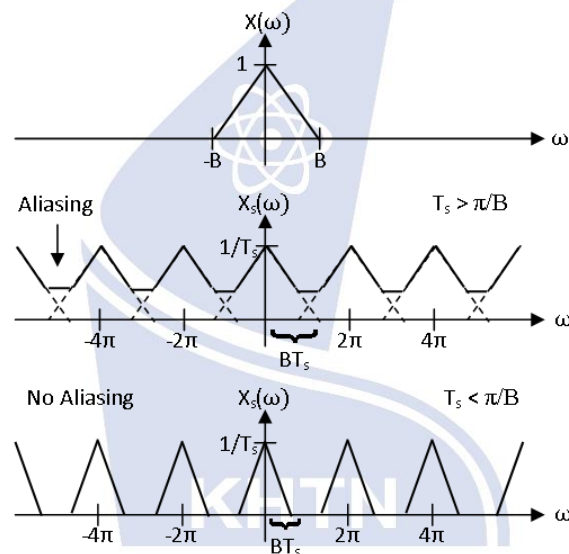


Fig 2.5 The Spectrum of a bandlimited signal is shown

## 2.4 Ngăn ngừa Aliasing

- Tăng tần số lấy mẫu
- Dùng bộ lọc thông thấp để loại bỏ thành phần tần số cao hơn tần số Nyquist.

### 3. Khôi phục tín hiệu (Reconstruction)

Tín hiệu băng tần  $x(t)$  có thể được phục hồi từ các mẫu của nó nếu tốc độ lấy mẫu là tốc độ Nyquist. Việc khôi phục này được thực hiện bằng cách truyền tín hiệu được lấy mẫu qua bộ lọc thông thấp lý tưởng có băng thông  $D$  và độ lợi  $T$ .

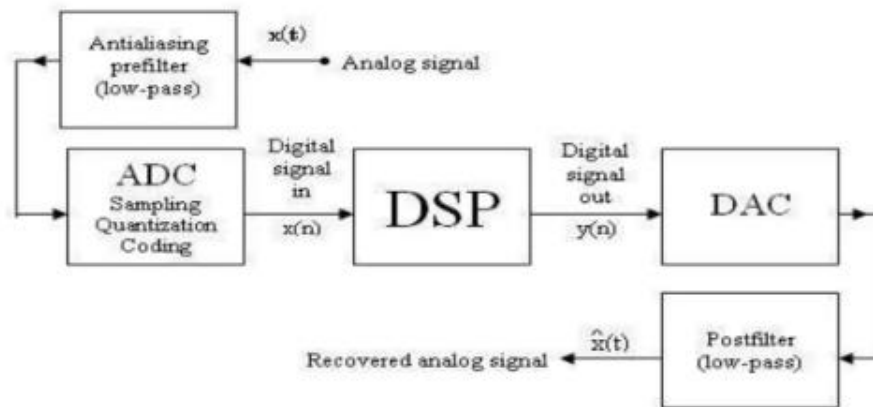


Fig 2.6 Block Diagram Reconstruction in The Frequency Domain to Find DTFT

- Các mẫu được chuyển thành chuỗi xung:

$$\sum_{n=-\infty}^{\infty} x(n) \delta(t - nTs) = \dots + x(-1) + \delta(n + Ts)$$

- Chuỗi xung được lọc qua bộ lọc thông thấp lý tưởng được giới hạn băng tần  $[-Fs/2, Fs/2]$ :

$$X_a(t) = \sum_{n=-\infty}^{\infty} x(n) \text{sinc}[Fs(t - nTs)]$$

$\text{Sinc}(x) = \frac{\sin(\pi x)}{\pi x}$  là một hàm nội suy bắt nguồn từ một đường truyền thấp lý tưởng của bộ lọc khôi phục. Tuy nhiên, vì bộ lọc thông thấp lý tưởng không thể được tìm trực tiếp nên, sẽ thường ước tính bộ lọc thông thấp lý tưởng bằng các phương pháp sau:

#### 3.1 Zero-order-hold (ZOH) interpolation:

Trong phép nội suy này, một giá trị mẫu nhất định được giữ trong khoảng thời gian lấy mẫu cho đến khi nhận được mẫu tiếp theo.

$$X_a(t) = x(n) \quad , nTs \leq t < (n+1)Ts$$



Có thể thu được bằng cách lọc chuỗi xung thông qua bộ lọc nội suy của dạng :

$$h_0(t) = \begin{cases} 1 & 0 \leq t < T_s \\ 0 & \text{khác} \end{cases} = \text{rect}\left(\frac{t - T/2}{T}\right)$$

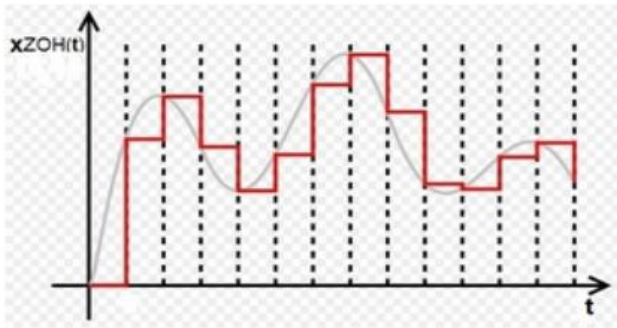


Fig 2.7 ZOH Reconstruct Signal

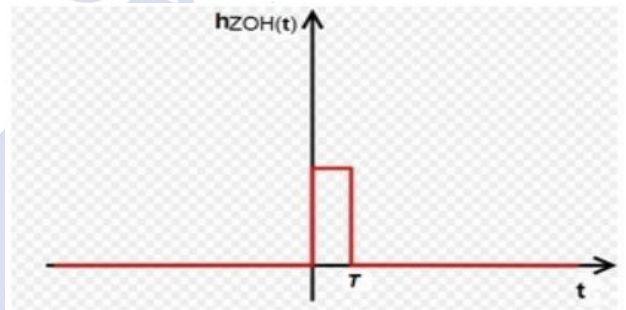
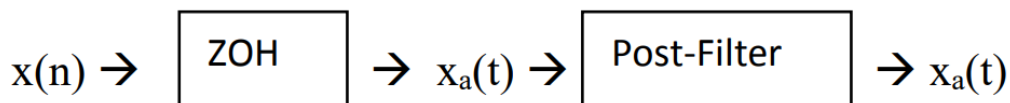


Fig 2.8 ZOH Impulse Response

Đây là một xung hình chữ nhật. Tín hiệu thu được là hằng số từng đoạn (bậc thang) dạng sóng yêu cầu bộ lọc sau tương tự được thiết kế phù hợp để có độ chính xác tái tạo dạng sóng.



### 3.2 First-order-hold (FOH) interpolation:

Trong trường hợp này các mẫu liên kế được nối bởi những đường thẳng. Điều này có thể đạt được bằng cách lọc chuỗi xung thông qua :

$$h_1(t) = \begin{cases} 1 + \frac{1}{T_s} & 0 \leq t < T_s \\ 1 - \frac{1}{T_s} & T_s \leq t < 2T_s \\ 0 & \text{khác} \end{cases}$$

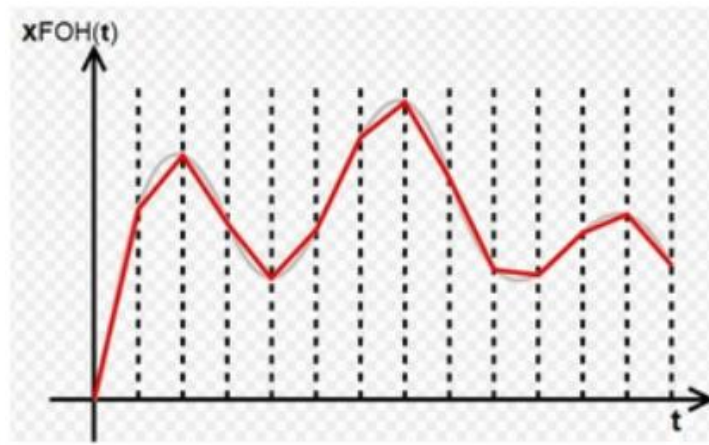


Fig 2.9 Reconstructed The Signal using Interpolated Line

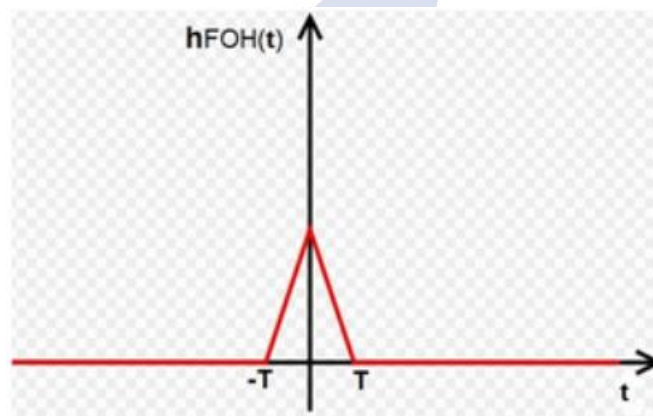


Fig 2.10 Impulse Response Caused By First-Order-Hold (FOH)

Như thể hiện trong hình trên, giá trị của tín hiệu nội suy bất cứ lúc nào phụ thuộc vào giá trị của mẫu trước và giá trị của mẫu tiếp theo và điều này không thể thực hiện được trong thời gian thực vì giá trị của mẫu tiếp theo không được biết trong thời gian thực. Nhưng nếu chúng ta trì hoãn tín hiệu được tái tạo một khoảng thời gian mẫu  $T$ , thì chúng ta có thể làm cho quá trình tái tạo trở thành nhân quả và tín hiệu được tái tạo sẽ xuất hiện như Fig 2.11.



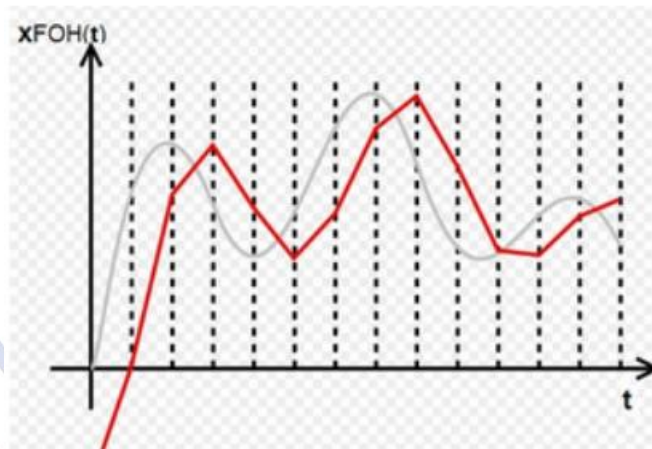
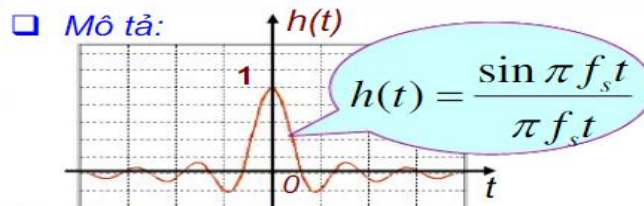


Fig 2.11 The Line Reconstructed Signal is Delayed By One Time Sample

Phép nội suy này có thể được thực hiện bằng cách tuân theo việc giữ thứ tự 0 bằng cách giữ thứ tự 0 giống hệt nhau. Điều này có nghĩa là đáp ứng xung của bộ lọc tái tạo tín hiệu như vậy sẽ là tích chập của đáp ứng xung giữ bậc 0 với chính nó :

$$h(t) = \text{rect}\left(\frac{t-T/2}{T}\right) * \text{rect}\left(\frac{t-T/2}{T}\right) = \text{tri}\left(\frac{t-T}{T}\right)$$



□ Quá trình khôi phục:

$$y_a(t) = \sum_{n=-\infty}^{\infty} y(nT_s) \frac{\sin \pi f_s (t - nT_s)}{\pi f_s (t - nT_s)}$$

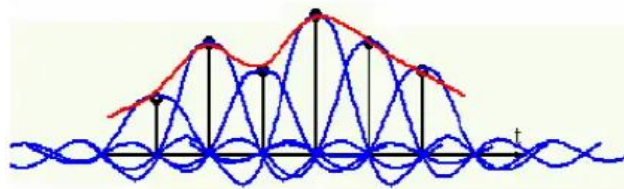
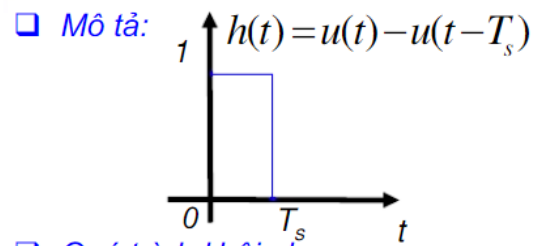


Fig 2.12 Describes The Signal Reconstructed Process In The Time Domain

### 3.3 Khôi phục hình thang :



□ Quá trình khôi phục:

$$y_a(t) = \sum_{n=-\infty}^{\infty} y(nT_s)[u(t - nT_s) - u(t - nT_s - T_s)]$$

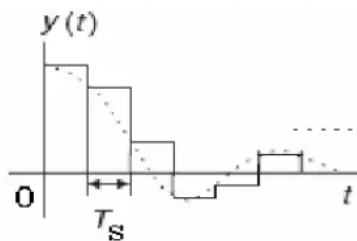


Fig 2.13.1 The Restructed Process of The Trapezoidal Restorer in the Time Domain

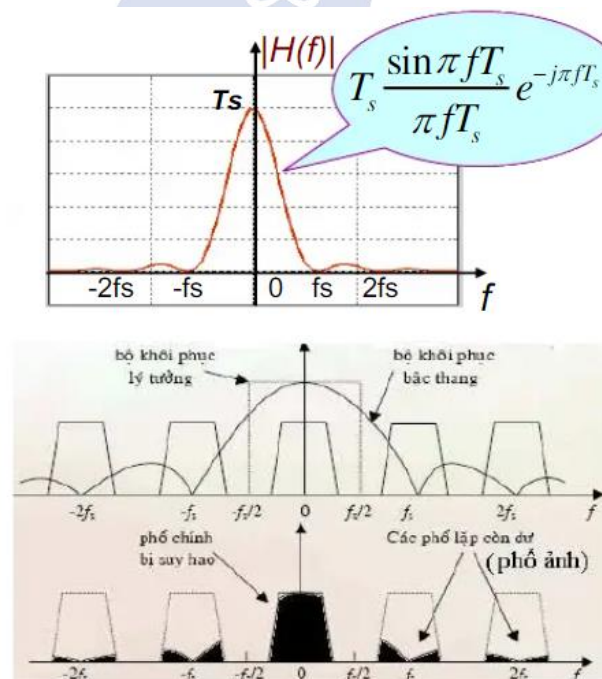
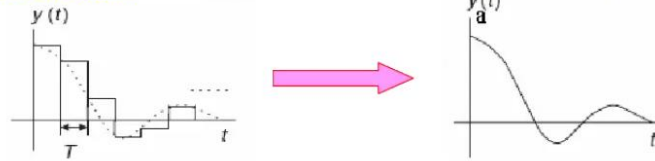


Fig 2.13.2 The Restructed Process of The Trapezoidal Restorer in the Time Frequency

Bộ hậu lọc: là bộ lọc thông thấp, nằm ngay sau bộ khôi phục hình thang, dùng để loại bỏ các thành phần phổ ảnh còn sót lại sau bộ lọc bậc thang. Từ đó nhận thấy bộ khôi phục lí tưởng có thể bao gồm bộ khôi phục hình thang + bộ hậu lọc.

❖ Miền thời gian:



❖ Miền tần số

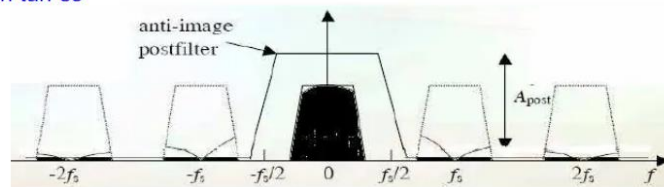


Fig 2.13.3 Result of the Restructed Process

## CHƯƠNG 3: MÔ PHỎNG MATLAB VÀ JUPYTER NOTEBOOK

### 3.1.1 Mô phỏng Matlab

a. Mô phỏng lấy mẫu tín hiệu với tín hiệu vào :

$$x_a(t) = 5\sin(20\pi t), 0 \leq t \leq 1.$$

$$f_{\max} = 20\pi / 2\pi = 10 \text{ Hz}$$

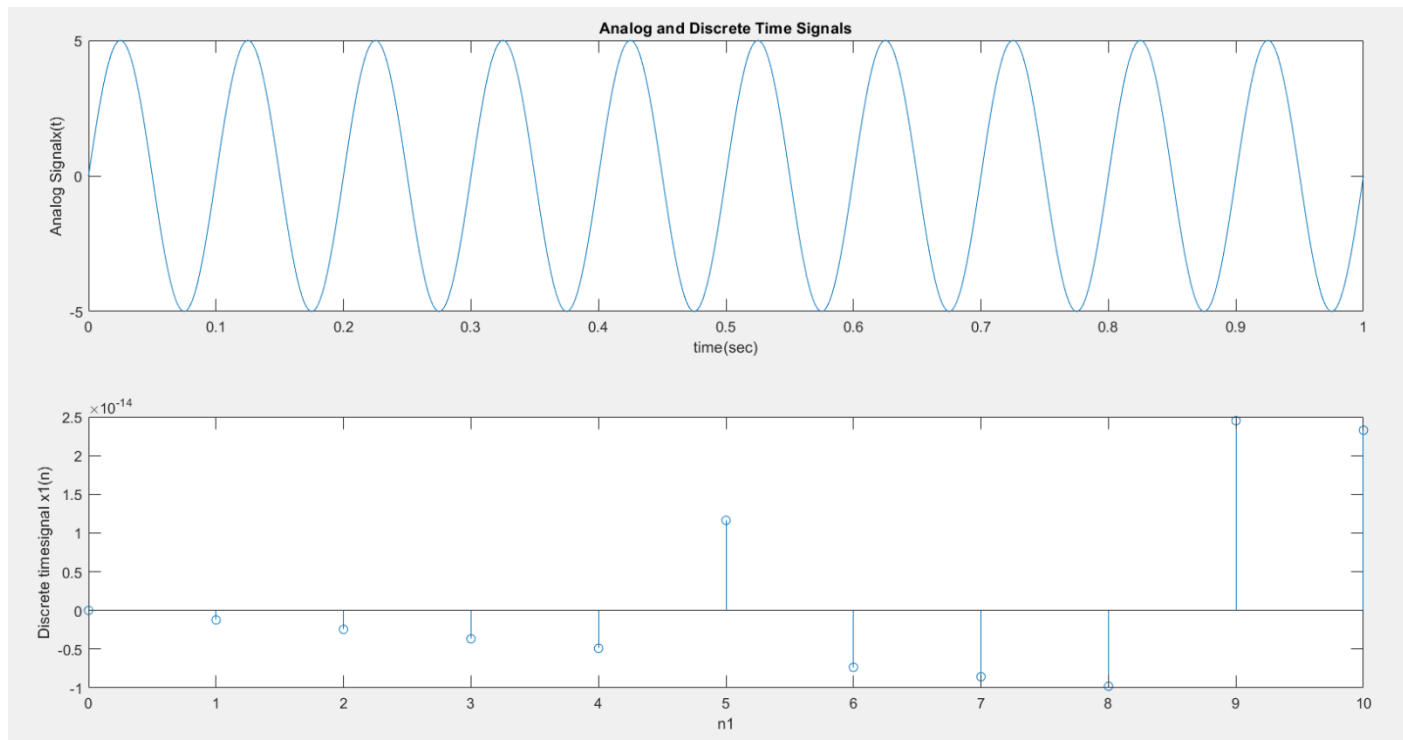
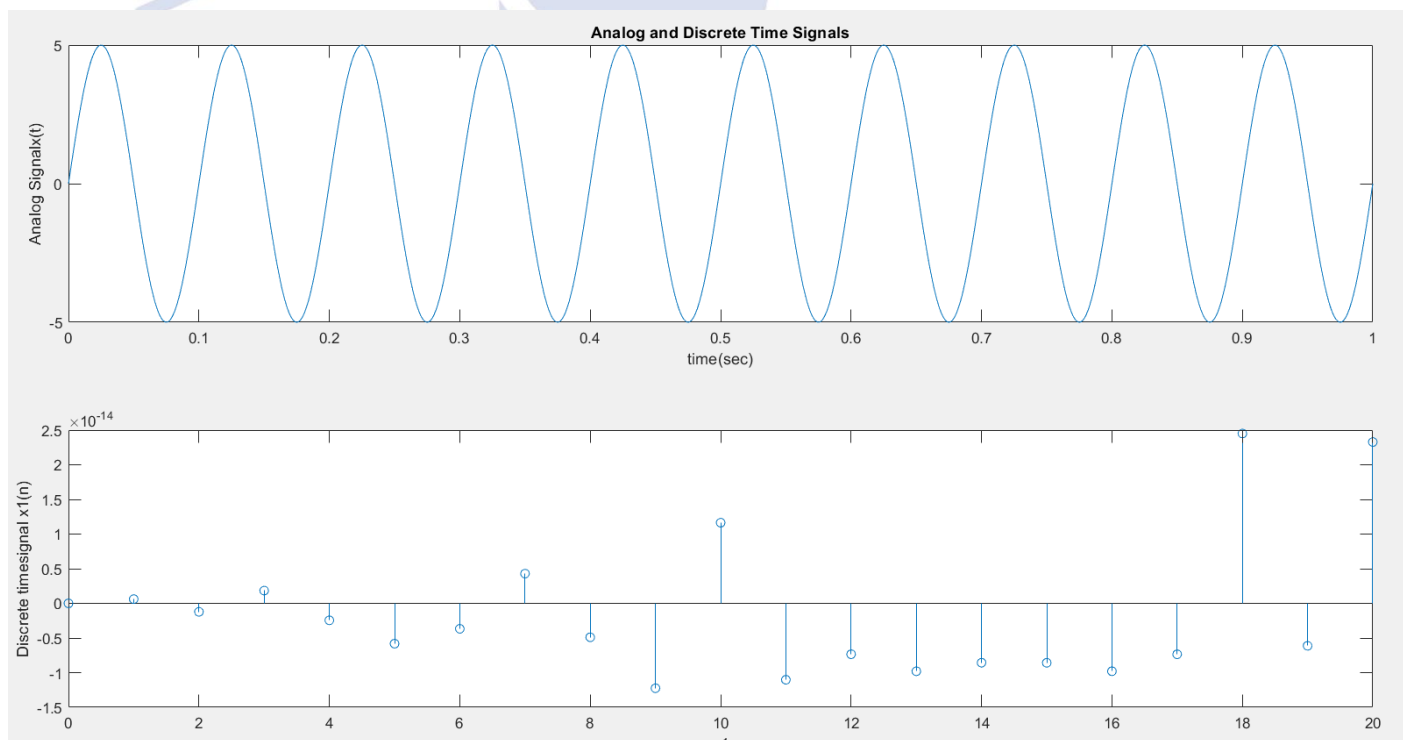
Theo định lý lấy mẫu Nyquist :  $f_s \geq 2f_{\max} \Leftrightarrow f_s \geq 20 \text{ Hz}$

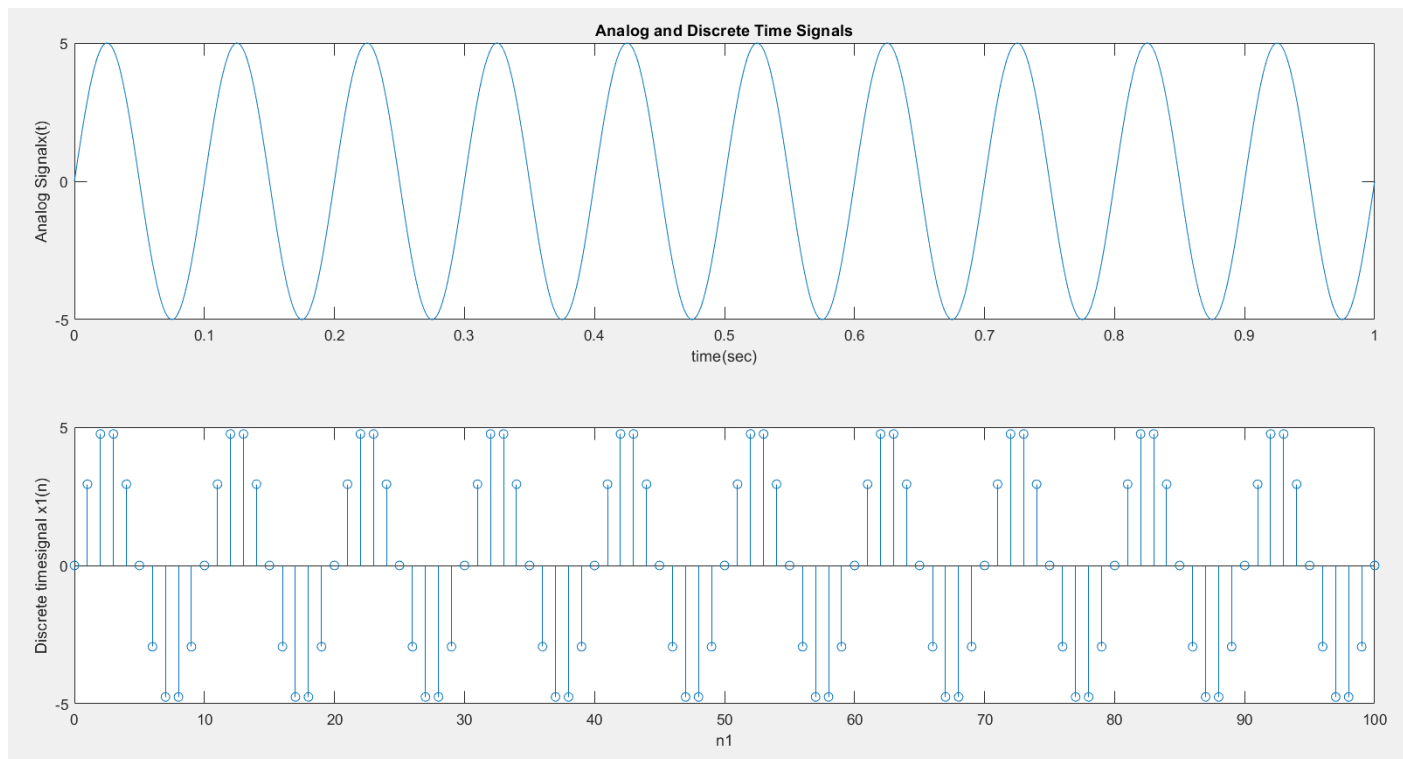
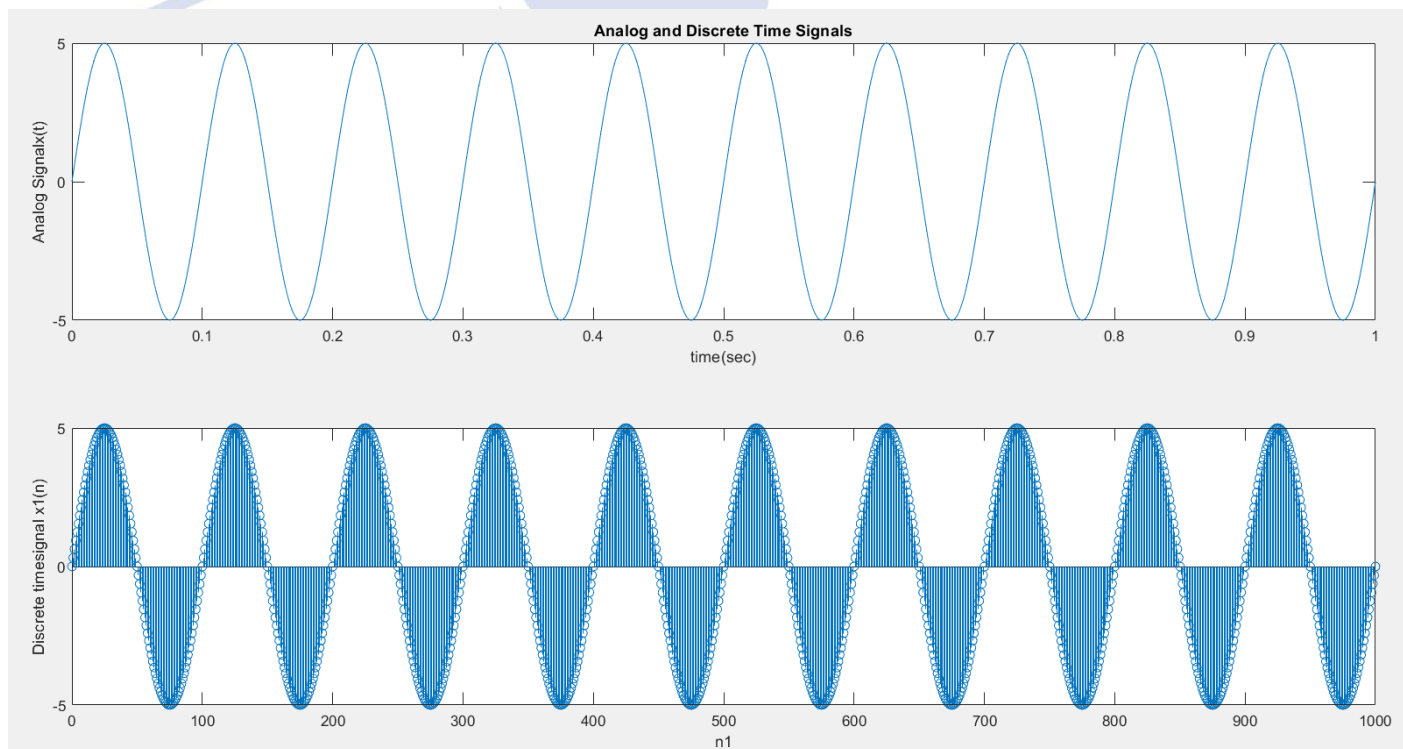
**Code:**

```
fs = input('Hay nhap fs : ');

tmin = 0;
tmax = 1;
t = tmin: 0.001: tmax;
xa = 5*sin(20*pi*t);
Ts1 = 1/fs;
n1 = tmin/Ts1:tmax/Ts1;
x1 = 5*sin(20*pi*n1*Ts1);

figure(5)
subplot(2,1,1)
plot(t,xa)
title('Analog and Discrete Time Signals')
xlabel('time(sec)')
ylabel('Analog Signal x(t)')
subplot(2,1,2)
stem(n1,x1)
xlabel('n1')
ylabel('Discrete timesignal x1(n)')
```

Fig 3.1.1 Khi  $f_s = 10\text{Hz}$ Fig 3.1.2 Khi  $f_s = 20\text{Hz}$

Fig 3.1.3 Khi  $f_s = 100\text{Hz}$ Fig 3.1.4 Khi  $f_s = 1000\text{Hz}$

**Nhận xét:**

- Tần số lấy mẫu càng lớn thì tín hiệu khôi phục càng giống với tín hiệu gốc.

**b. Mô phỏng khôi phục tín hiệu trên****Code:**

```

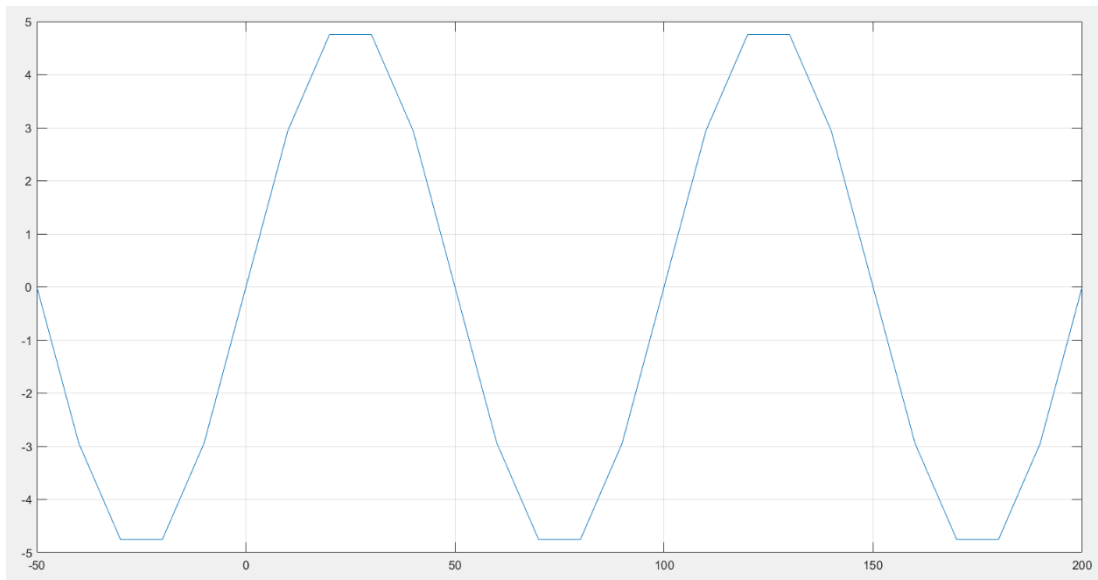
Ts = 0.01;
Fs = 1/Ts;
n = -5:1:20 ;
nTs = n*Ts;
x = 5*sin(20*pi.*nTs);

Dt = 0.00005;
t = -0.005:Dt:0.005;
xa = x*sinc(Fs*(ones(length(n),1)*t - nTs'*ones(1,length(t))));
%xa = sin(20 * pi * t);

subplot(1,1,1);
plot(t*1000,xa)
xlabel('time in (msec)');
ylabel('xa(t)')
plot (n*Ts*1000,x);
grid on
hold off

%Sai so toi da giua tin hieu tương tự được tái tạo và tín hiệu tương tự thực tế
error = max(abs(xa - sin(20*pi*t)));

```



*Fig 3.2 Kết quả khôi phục tín hiệu khi chọn  $f_s=100\text{Hz}$*

→ Sai số tối đa giữa tín hiệu tương tự được tái tạo và tín hiệu tương tự thực tế là 1.1766

### 3.1.2 Mô phỏng bằng Simulink

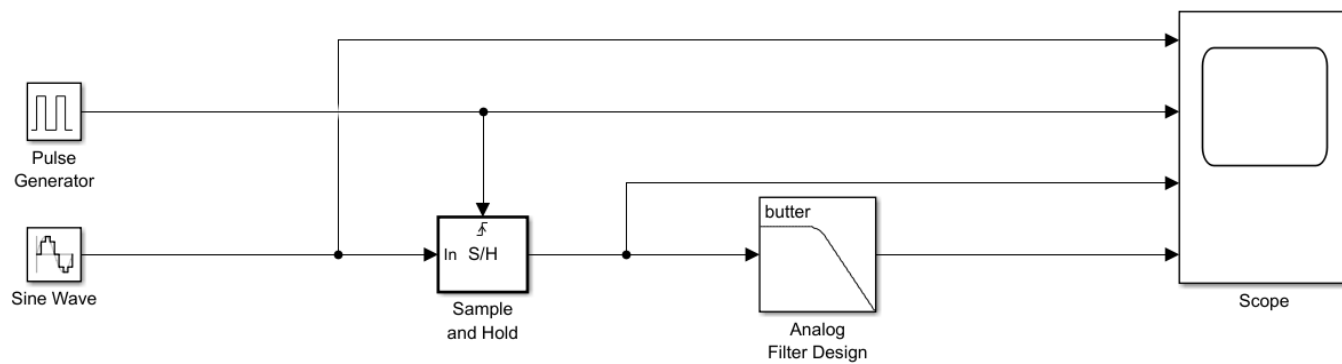


Fig 3.3.1 Sơ đồ khối quá trình lấy mẫu và khôi phục

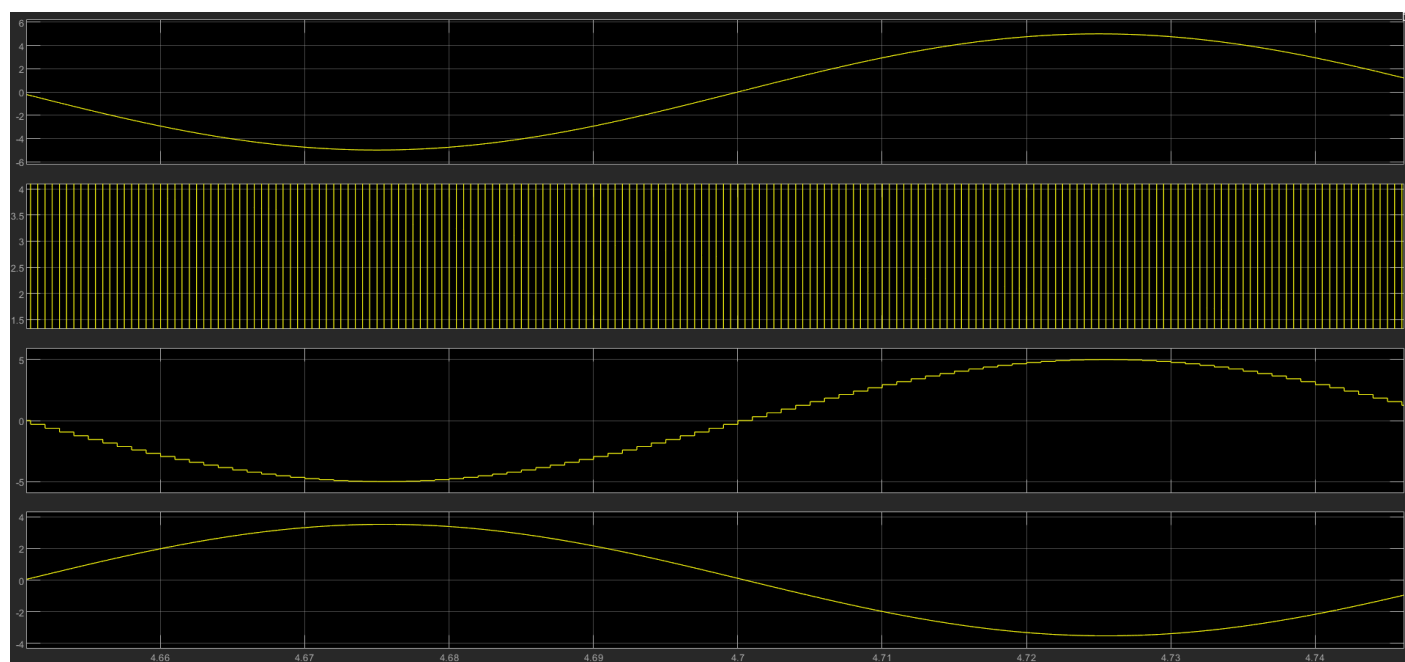


Fig 3.3.2 Toàn bộ tín hiệu quan sát được qua Scope



### 3.2 Mô phỏng Jupyter Notebook

```
import matplotlib.pyplot as plt # Thư viện  
  
from scipy import signal  
  
import numpy as np
```

Tín hiệu có dạng:

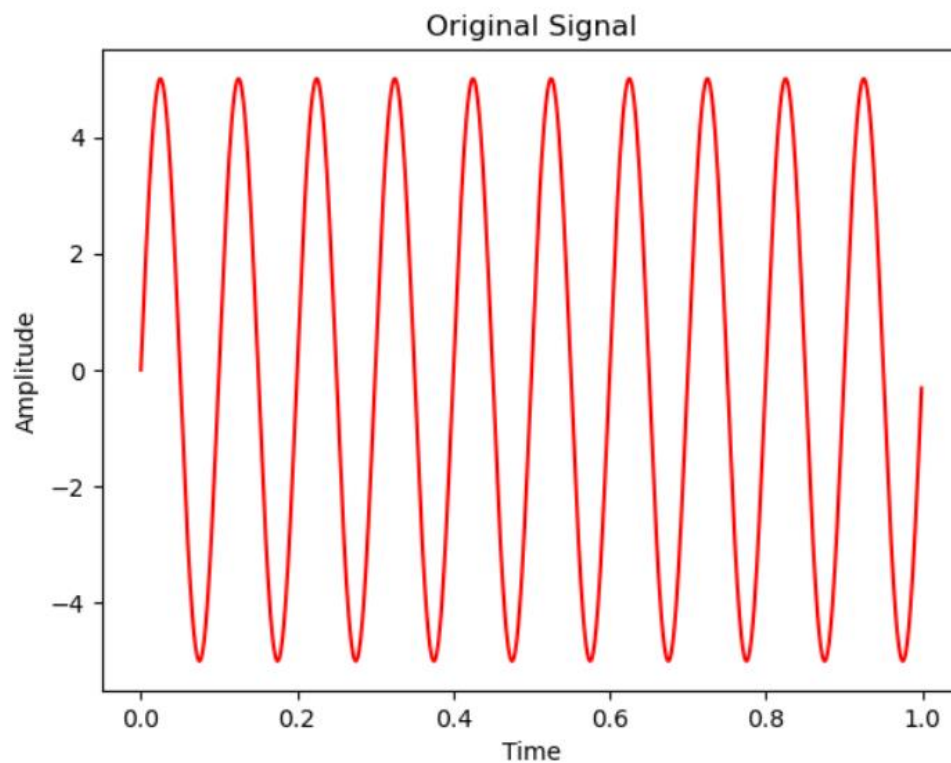
```
x = 5*sin(2*np.pi*10*t)
```

→  $f_{\max} = 20\pi/2\pi = 10$

Theo định lý lấy mẫu nyquist:  $f_s \geq 2f_{\max} = 20\text{Hz}$

**Code:**

```
t = np.linspace(0, 1, 1000, endpoint=False)  
  
x = 5*np.sin(2*np.pi*10*t)  
  
'''  
Tín hiệu gốc  
'''  
  
plt.plot(t, x, 'r')  
plt.title('Original Signal')  
plt.xlabel('Time')  
plt.ylabel('Amplitude')  
plt.show()
```

*Fig 3.4.1: Tín hiệu gốc*

```
'''  
Lấy mẫu từng fs  
'''  
  
fs1 = 10 # Trường hợp fs < 2fmax --> aliasing  
fs2 = 20 # fs = 2fmax  
fs3 = 500 # fs > 2fmax  
  
ts1 = 1/fs1  
ts2 = 1/fs2  
ts3 = 1/fs3
```

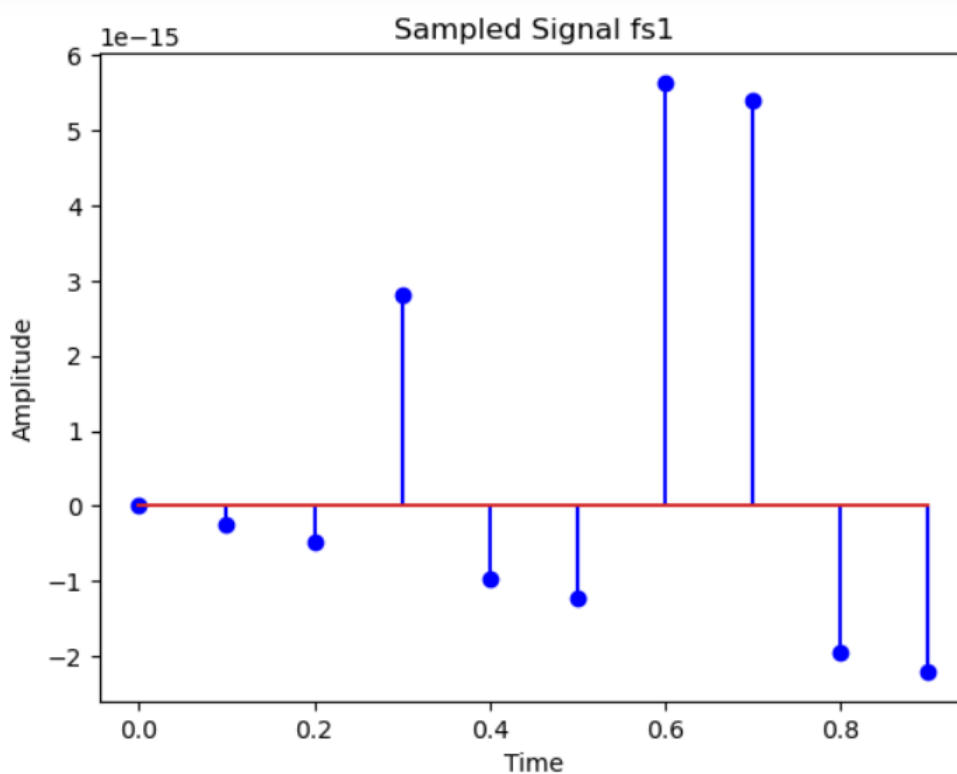
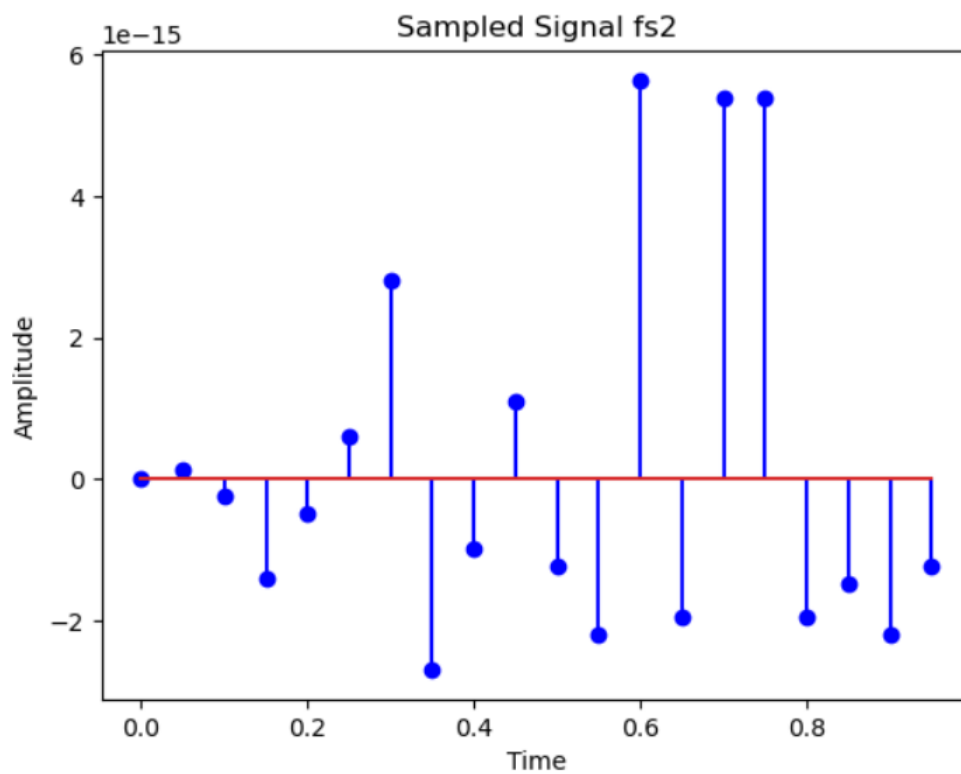
```
t_sampled1 = np.arange(0, 1, ts1)
t_sampled2 = np.arange(0, 1, ts2)
t_sampled3 = np.arange(0, 1, ts3)

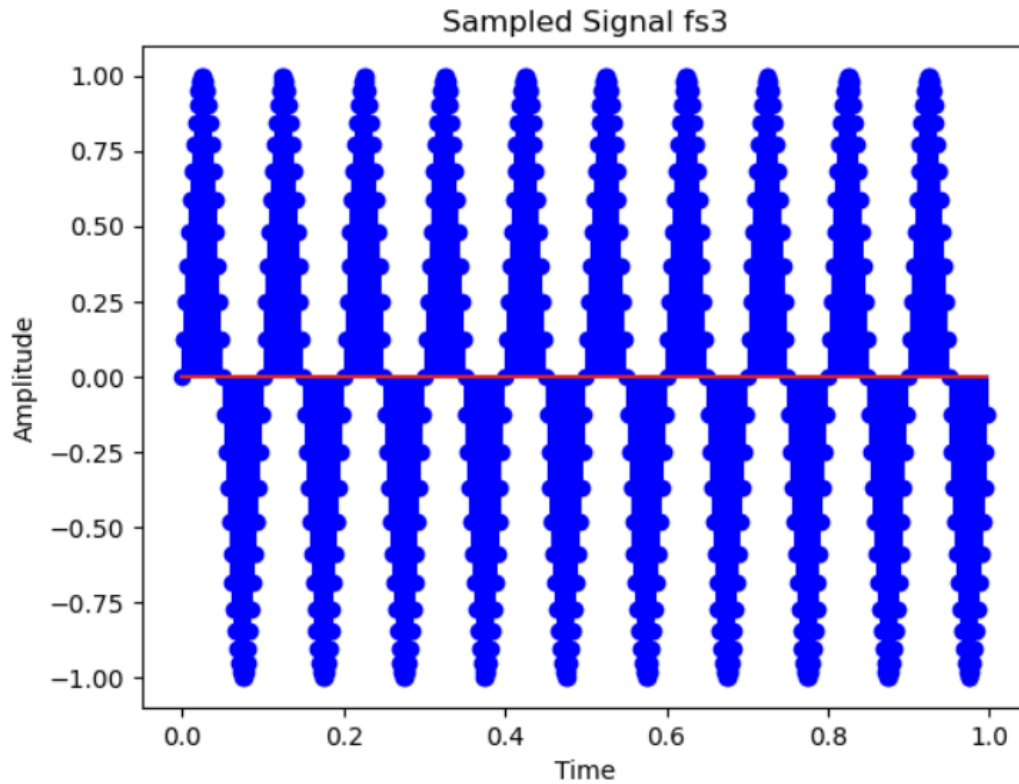
x_sampled1 = np.sin(2*np.pi*10*t_sampled1)
x_sampled2 = np.sin(2*np.pi*10*t_sampled2)
x_sampled3 = np.sin(2*np.pi*10*t_sampled3)

plt.stem(t_sampled1, x_sampled1, 'b')
plt.title('Sampled Signal fs1')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.stem(t_sampled2, x_sampled2, 'b')
plt.title('Sampled Signal fs2')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.stem(t_sampled3, x_sampled3, 'b')
plt.title('Sampled Signal fs3')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```

*Fig 3.4.2: Tín hiệu lấy mẫu của fs1**Fig 3.4.3: Tín hiệu lấy mẫu của fs2*



*Fig 3.4.4: Tín hiệu lấy mẫu của fs3*

```
'''
Khôi phục tín hiệu của từng fs
'''
x_reconstructed_1 = np.interp(t, t_sampled1, x_sampled1)

plt.plot(t, x_reconstructed_1, 'g')
plt.title('Reconstructed Signal fs1')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

x_reconstructed_2 = np.interp(t, t_sampled2, x_sampled2)
```

```
plt.plot(t, x_reconstructed_2, 'g')
plt.title('Reconstructed Signal fs2')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

x_reconstructed_3 = np.interp(t, t_sampled3, x_sampled3)

plt.plot(t, x_reconstructed_3, 'g')
plt.title('Reconstructed Signal fs3')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```

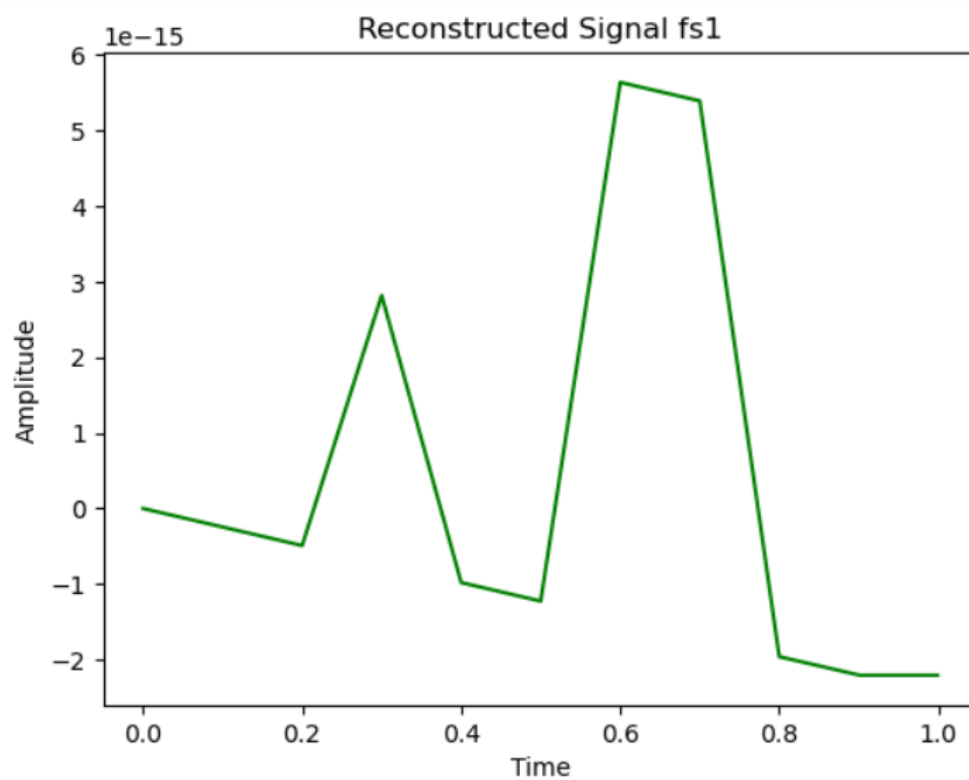


Fig 3.4.5: Tín hiệu khôi phục của  $fs1$

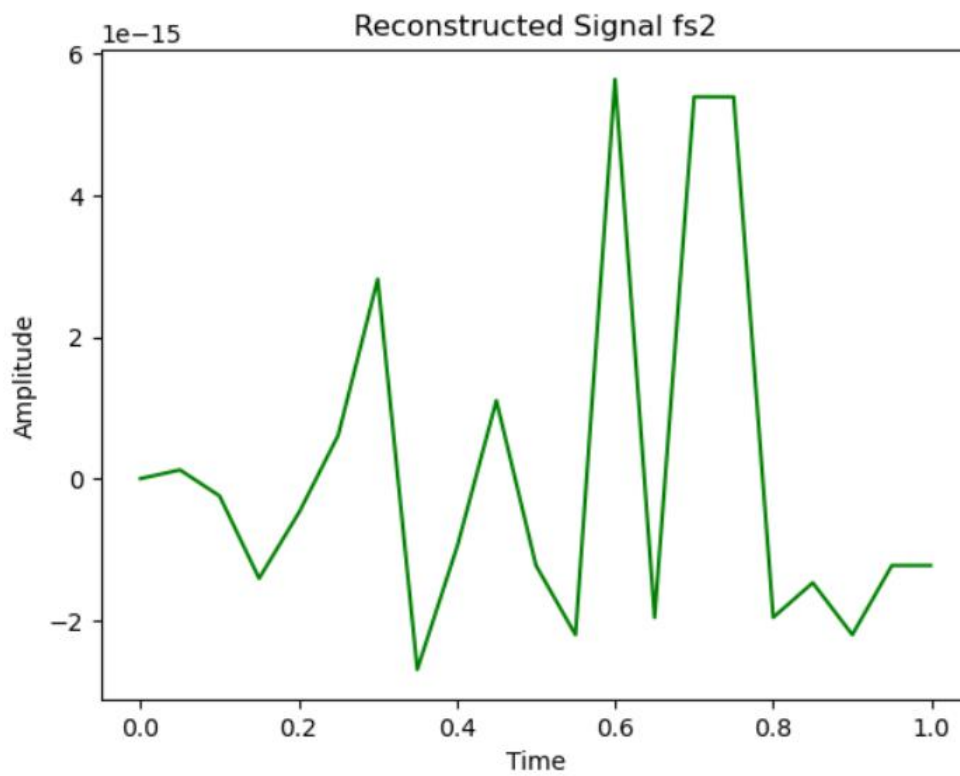
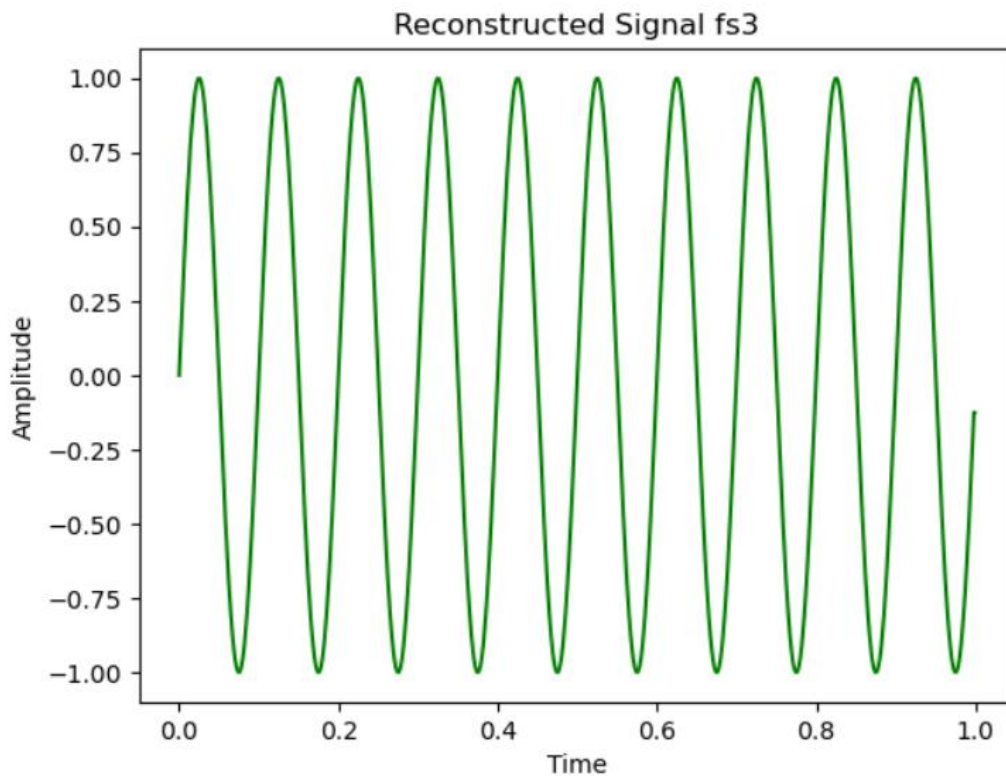


Fig 3.4.6: Tín hiệu khôi phục của fs2





Hình 4.1.7: Tín hiệu khôi phục của  $fs3$

Nhận xét:

- Tần số lấy mẫu càng lớn thì tín hiệu khôi phục càng giống với tín hiệu gốc

## TÀI LIỆU THAM KHẢO

[1] Slide bài giảng môn Xử Lý Tín Hiệu Số

[2] JNAGA SAI. (2016-2017). *Sampling and Reconstruction of Analog Signals with Aliasing*. Slideshare. Truy cập vào 10/12/2023, từ <https://www.slideshare.net/slideshow/sampling-and-reconstruction-of-signal-using-aliasing/68251091>