西北工业大学 《信号与系统》实验报告

学	院 :	软件学院
学	号:	2021302853
姓	名:	张苏宇
专	<u> </u>	软件工程
实验	时间 :	2023年12月14日
实验	地点:	启翔楼 264
指导	教师:	柳艾飞、汪彦婷

西北工业大学

2023 年 12 月

一、实验目的

- Ø 掌握抽样定理,验证抽样定理;
- Ø 掌握利用Matlab完成信号抽样的方法,并对抽样信号的频谱进行分析;
- Ø 了解运用Matlab对抽样信信号进行恢复的方法。

二、实验报告要求

1. 提交:实验报告一份, PDF格式, 其他格式拒收。

实验报告中需要包括:

- a) 若题目要求理论结果, 报告中需要给出理论结果。
- b) 结果图;图中需要有适当的标识。
- c) 源代码。源代码中要有合适的注释。
- d) 实验体会和感悟。
 - 2. 提交实验报告规则:

2023年12月17日21点之前将实验报告提交到坚果云

一班实验三提交链接: https://send2me.cn/fMgZAygK/TpSFwww7gnHHeA

二班实验三提交链接: https://send2me.cn/9btDbArW/QJC S42AQ_IXng

文件名命名规则:课堂号-学号-姓名-第几次实验。

三、实验设备 (环境)

操作系统: Windows 10

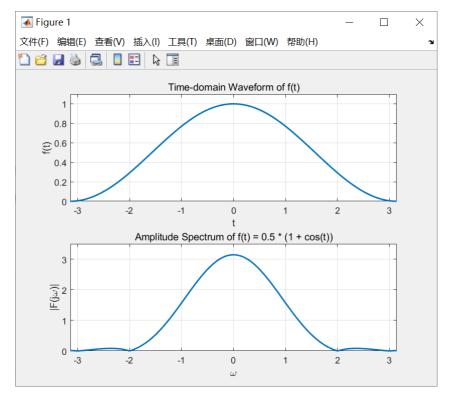
编程软件: Matlab R2023b

四、实验内容

Ø 实验1: 抽样定理验证实验

已知连续信号为 $f(t) = 0.5(1 + cost), -\pi \le t \le \pi$

(1) 绘制f(t)时域波形和频谱;

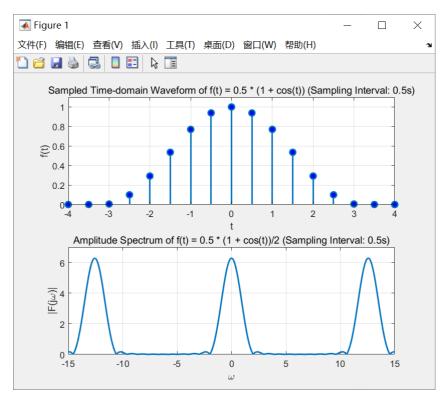


```
1
    function plotSignalAndSpectrum()
 2
        % Define time vector
 3
        t = -pi:0.01:pi;
 4
 5
        % Define the signal function
        f = ((1 + cos(t)) / 2) .* (stepfun(t, -pi) - stepfun(t, pi));
 6
 7
 8
        % Plot time-domain waveform
9
        figure;
10
        subplot(2, 1, 1);
        plot(t, f, 'LineWidth', 1.5);
11
12
        title('Time-domain Waveform of f(t)');
13
        grid on;
        xlabel('t');
14
        xlim([-pi, pi]);
15
        ylabel('f(t)');
16
17
        ylim([0, 1.1]);
18
        % Compute and plot the amplitude spectrum
19
20
        omega1 = -pi;
21
        omega2 = pi;
22
        K = 4000;
23
        OMEGA = omega2 - omega1;
24
        delta_omega = OMEGA / K;
25
        omega = omega1:delta_omega:omega2;
26
        F = 0.01 * (f * exp(-1i * t' * omega));
27
28
        subplot(2, 1, 2);
29
        plot(omega, abs(F), 'Linewidth', 1.5);
30
        title('Amplitude Spectrum of f(t) = 0.5 * (1 + cos(t))');
31
32
        grid on;
        xlabel('\omega');
33
34
        xlim([-pi, pi]);
```

```
35  ylabel('|F(j\omega)|');
36  ylim([0, 3.5]);
37  end
38
```

(2) 分别绘制抽样间隔为0.5s、1s、2s时的抽样信号的时域波形和频谱;

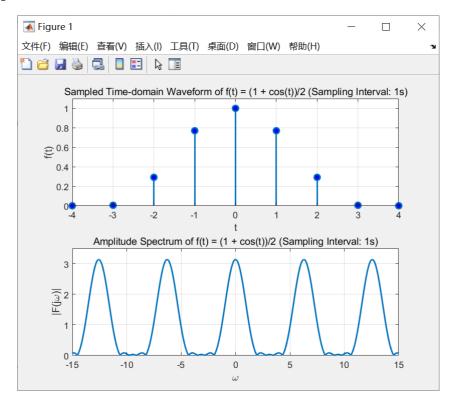
①抽样间隔为 0.5s:



```
1
    function plotSampledSignalAndSpectrum()
 2
        % Define sampling interval
 3
        Ts = 0.5;
 4
        % Define time vector
 5
 6
        ts = -20:Ts:20;
 7
 8
        % Define the sampled signal
        fs = ((1 + cos(ts)) / 2) .* (heaviside(ts + pi) - heaviside(ts - pi));
 9
10
11
        % Plot time-domain waveform
        figure;
12
13
        subplot(2, 1, 1);
        stem(ts, fs, 'MarkerFaceColor', 'b', 'LineWidth', 1.5);
14
        title('Sampled Time-domain waveform of f(t) = 0.5 * (1 + cos(t))
15
    (Sampling Interval: 0.5s)');
16
        grid on;
17
        xlabel('t');
18
        xlim([-4, 4]);
19
        ylabel('f(t)');
20
        ylim([0, 1.1]);
21
22
        % Compute and plot the amplitude spectrum
23
        omega1 = -8 * pi;
        omega2 = 8 * pi;
24
25
        K = 4000;
```

```
26
        OMEGA = omega2 - omega1;
27
        delta_omega = OMEGA / K;
28
        omega = omega1:delta_omega:omega2;
29
        % Calculate Fourier series coefficients
30
        Fs = Ts * (fs * exp(-1i * ts' * omega));
31
32
33
        subplot(2, 1, 2);
34
        plot(omega, abs(2 * Fs), 'LineWidth', 1.5);
35
        title('Amplitude Spectrum of f(t) = 0.5 * (1 + \cos(t))/2 (Sampling
    Interval: 0.5s)');
36
        grid on;
37
        xlabel('\omega');
38
        xlim([-15, 15]);
        ylabel('|F(j\omega)|');
39
40
        ylim([0, 7]);
    end
41
42
```

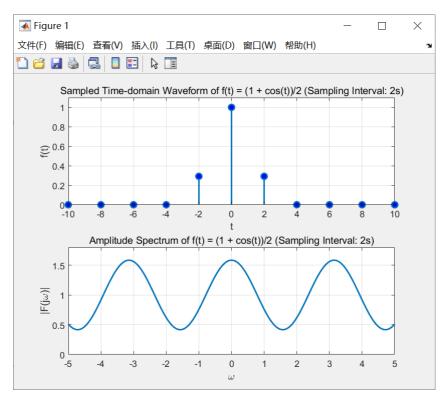
②抽样间隔为 1s:



```
1
    function plotSampledSignalAndSpectrum()
 2
        % Parameters
 3
        Ts = 1;
 4
        tRange = -20:Ts:20;
 5
        omegaRange = -8*pi:0.001:8*pi;
 6
 7
        % Define the sampled signal
        fs = ((1 + cos(tRange)) / 2) .* (heaviside(tRange + pi) -
 8
    heaviside(tRange - pi));
9
        % Plot time-domain waveform
10
11
        figure;
12
        subplot(2, 1, 1);
```

```
stem(tRange, fs, 'MarkerFaceColor', 'b', 'LineWidth', 1.5);
13
14
        title('Sampled Time-domain Waveform of f(t) = (1 + cos(t))/2 (Sampling
    Interval: 1s)');
        grid on;
15
16
        xlabel('t');
17
        xlim([-4, 4]);
        ylabel('f(t)');
18
19
        ylim([0, 1.1]);
20
21
        % Compute and plot the amplitude spectrum
        omega1 = -8 * pi;
22
        omega2 = 8 * pi;
23
24
        K = 4000;
25
        delta_omega = (omega2 - omega1) / K;
        omega = omega1:delta_omega:omega2;
26
27
28
        % Calculate Fourier series coefficients
29
        Fs = Ts * (fs * exp(-1i * tRange' * omega));
30
31
        subplot(2, 1, 2);
32
        plot(omega, abs(Fs), 'LineWidth', 1.5);
        title('Amplitude Spectrum of f(t) = (1 + \cos(t))/2 (Sampling Interval:
33
    1s)');
34
        grid on;
35
        xlabel('\omega');
36
        xlim([-15, 15]);
37
        ylabel('|F(j\omega)|');
38
        ylim([0, 3.5]);
39
    end
40
```

③抽样间隔为 2s:



```
% Parameters
3
        Ts = 2;
4
        tRange = -20:Ts:20;
 5
        omegaRange = -10*pi:0.001:10*pi;
 6
7
        % Define the sampled signal
        fs = ((1 + cos(tRange)) / 2) .* (heaviside(tRange + pi) -
8
    heaviside(tRange - pi));
9
        % Plot time-domain waveform
10
11
        figure;
12
        subplot(2, 1, 1);
13
        stem(tRange, fs, 'MarkerFaceColor', 'b', 'LineWidth', 1.5);
        title('Sampled Time-domain Waveform of f(t) = (1 + cos(t))/2 (Sampling
14
    Interval: 2s)');
15
        grid on;
        xlabel('t');
16
17
        xlim([-10, 10]);
18
        ylabel('f(t)');
19
        ylim([0, 1.1]);
20
        % Compute and plot the amplitude spectrum
21
22
        omega1 = -10 * pi;
23
        omega2 = 10 * pi;
24
        K = 4000;
25
        delta_omega = (omega2 - omega1) / K;
26
        omega = omega1:delta_omega:omega2;
27
28
        % Calculate Fourier series coefficients
        Fs = Ts * (fs * exp(-1i * tRange' * omega));
29
30
31
        subplot(2, 1, 2);
32
        plot(omega, abs(Fs / 2), 'LineWidth', 1.5);
33
        title('Amplitude Spectrum of f(t) = (1 + \cos(t))/2 (Sampling Interval:
    2s)');
34
        grid on;
35
        xlabel('\omega');
36
        xlim([-5, 5]);
37
        ylabel('|F(j\omega)|');
38
        ylim([0, 1.8]);
39
    end
40
```

(3)观察抽样信号的频谱混叠程度,验证抽样定理。注:抽样信号的幅度谱绘制三个周期即可。

抽样定理(Nyquist定理)是一个基本的信号处理原理,它指导着在进行信号抽样时应该选择多大的抽样率。抽样定理的核心观点是,为了避免频谱混叠(即采样导致的信号信息丢失),信号的抽样率必须至少是信号中最高频率的两倍。

```
原信号的 \omega_m=2, \omega_s=rac{2\pi}{T}
```

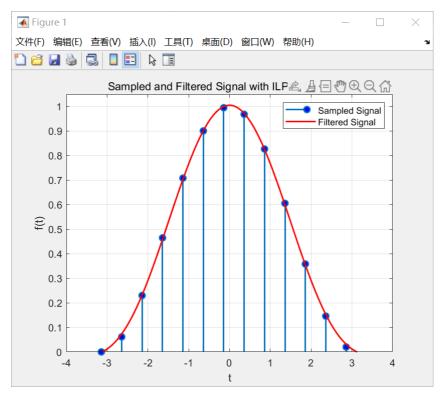
当抽样间隔抽样间隔为 0.5s 时, $\omega_s=4\pi$,抽样间隔为 1s 时, $\omega_s=2\pi$,这两种情况时 $\omega_s\geq 2\omega_m$,大于奈奎斯特抽样频率,相邻信号之间没发生混叠。

而当抽样间隔为 2s 时, $\omega_s \leq 2\omega_m$,小于奈奎斯特抽样频率,相邻信号之间发生混叠,间隔信号失真,满足抽样定理。

Ø 实验2: 信号恢复实验

- 2.1 对实验1中的信号,观察到 $\omega_m=2$ 。对于抽样之后的信号,采用截止频率为 $\omega_c=1.2\omega_m$ 的 ILPF进行信号恢复。
- (1) 画出三种抽样间隔下抽样信号通过ILPF后的信号时域波形图;

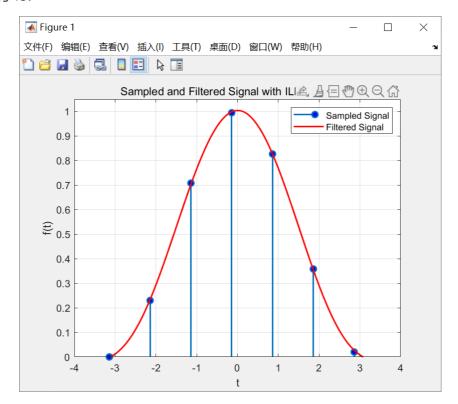
①抽样间隔为 0.5s:



```
1
    function plotSampledAndFilteredSignal()
 2
        % Sampling parameters
 3
        Ts = 0.5;
 4
        t_sampling = -pi:Ts:pi;
 5
        % Continuous time vector
 6
 7
        t_continuous = -pi:0.01:pi;
 8
        % Signal definition
 9
10
        f_{continuous} = 0.5 * (1 + cos(t_{sampling}));
11
        % Ideal Low Pass Filter
12
13
        Wc = 2.4;
        ILPF = sinc(Wc/pi * (t_continuous - t_sampling'));
14
15
        % Sampled and filtered signal
16
        F_filtered = Ts * (Wc/pi) * f_continuous * ILPF;
17
18
19
        % Plotting
20
        figure;
        stem(t_sampling, f_continuous, 'MarkerFaceColor', 'b', 'LineWidth',
21
    1.5);
22
        hold on;
23
        plot(t_continuous, F_filtered, 'r', 'LineWidth', 1.5);
```

```
24
        hold off;
25
26
        xlabel('t');
27
        ylabel('f(t)');
        ylim([0, 1.05]);
28
29
        title('Sampled and Filtered Signal with ILPF (Ts = 0.5s)');
30
        legend('Sampled Signal', 'Filtered Signal');
31
        grid on;
32
    end
33
```

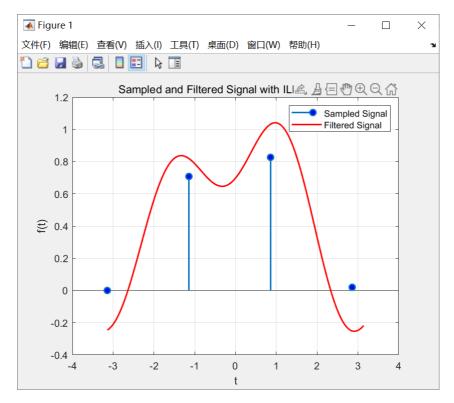
②抽样间隔为 1s:



```
1
    function plotSampledAndFilteredSignal()
 2
        % Sampling parameters
 3
        Ts = 1.0;
 4
        t_sampling = -pi:Ts:pi;
 5
 6
        % Continuous time vector
 7
        t_continuous = -pi:0.01:pi;
 8
9
        % Signal definition
10
        f_{continuous} = 0.5 * (1 + cos(t_{sampling}));
11
12
        % Ideal Low Pass Filter
13
        Wc = 2.4;
14
        ILPF = sinc(wc/pi * (t_continuous - t_sampling'));
15
        % Sampled and filtered signal
16
17
        F_filtered = Ts * (Wc/pi) * f_continuous * ILPF;
18
        % Plotting
19
20
        figure;
```

```
stem(t_sampling, f_continuous, 'MarkerFaceColor', 'b', 'LineWidth',
21
    1.5);
22
        hold on;
23
        plot(t_continuous, F_filtered, 'r', 'LineWidth', 1.5);
        hold off;
24
25
26
        xlabel('t');
27
        ylabel('f(t)');
28
        ylim([0, 1.05]);
29
        title('Sampled and Filtered Signal with ILPF (Ts = 1s)');
        legend('Sampled Signal', 'Filtered Signal');
30
        grid on;
31
32
    end
33
```

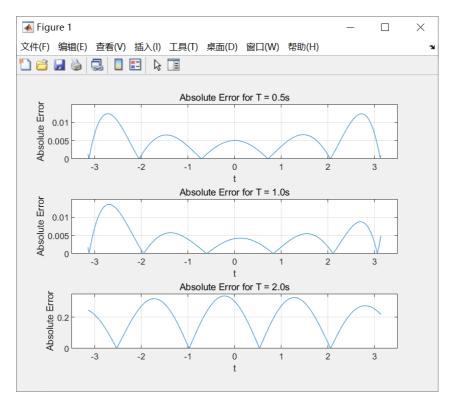
③抽样间隔为 2s:



```
1
    function plotSampledAndFilteredSignal()
 2
        % Sampling parameters
 3
        Ts = 2.0;
 4
        t_sampling = -pi:Ts:pi;
 5
 6
        % Continuous time vector
 7
        t_continuous = -pi:0.01:pi;
 8
9
        % Signal definition
10
        f_{continuous} = 0.5 * (1 + cos(t_{sampling}));
11
12
        % Ideal Low Pass Filter
13
        Wc = 2.4;
        ILPF = sinc(Wc/pi * (t_continuous - t_sampling'));
14
15
16
        % Sampled and filtered signal
17
        F_filtered = Ts * (Wc/pi) * f_continuous * ILPF;
```

```
18
19
        % Plotting
20
        figure;
        stem(t_sampling, f_continuous, 'MarkerFaceColor', 'b', 'LineWidth',
21
    1.5);
        hold on;
22
        plot(t_continuous, F_filtered, 'r', 'LineWidth', 1.5);
23
        hold off;
24
25
26
        xlabel('t');
        ylabel('f(t)');
27
        title('Sampled and Filtered Signal with ILPF (Ts = 2s)');
28
29
        legend('Sampled Signal', 'Filtered Signal');
30
        grid on;
    end
31
32
```

(2)绘制三种抽样间隔下的恢复信号与原信号的绝对误差图,观察并总结抽样间隔对于信号恢复过程的 影响。



由以上三张绝对误差图可知:

- ①当抽样间隔为 0.5s 或为 1s 时, $\omega_s \geq 2\omega_m$,还原出来的在理想情况下不失真,恢复信号与原信号相比,绝对误差较小;
- ②当抽样间隔为 2s 时, $\omega_s \leq 2\omega_m$, 还原出来的信号明显发生失真,采样以后的恢复信号与原信号相比,绝对误差较大。

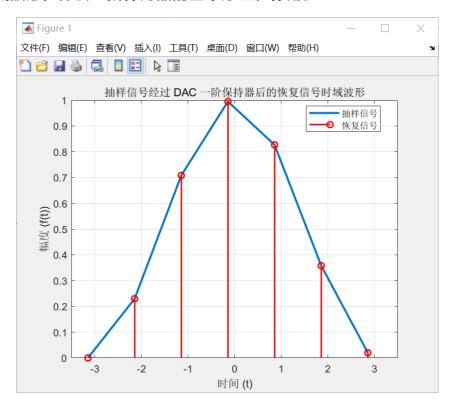
在信号恢复过程中选择的抽样间隔越小, ω_s 越大,信号恢复的效果越好,反之效果越差。选择恰当的抽样间隔才能既不必过于频繁的采样,又保证信号的失真在可容忍的范围内。

```
function plotAbsoluteErrorComparison()
% Sampling intervals
T1 = 0.5;
T2 = 1.0;
```

```
5
        T3 = 2.0;
 6
 7
        % Sample ranges
        t1 = -pi:T1:pi;
8
9
        t2 = -pi:T2:pi;
10
        t3 = -pi:T3:pi;
        t = -pi:0.01:pi;
11
12
        Wc = 2.4;
13
        % Original signal
14
        f = 0.5 * (1 + cos(t));
15
16
17
        % Sampled and reconstructed signals
        f1 = 0.5 * (1 + cos(t1));
18
        F1 = T1 * (Wc/pi) * f1 * sinc(Wc/pi * (ones(length(t1),1) * t - t1' *
19
    ones(1,length(t)));
20
        f2 = 0.5 * (1 + cos(t2));
21
        F2 = T2 * (Wc/pi) * f2 * sinc(Wc/pi * (ones(length(t2),1) * t - t2' *
22
    ones(1,length(t)));
23
        f3 = 0.5 * (1 + cos(t3));
24
25
        F3 = T3 * (Wc/pi) * f3 * sinc(Wc/pi * (ones(length(t3),1) * t - t3' *
    ones(1,length(t)));
26
27
        % Absolute errors
28
        error1 = abs(F1 - f);
29
        error2 = abs(F2 - f);
30
        error3 = abs(F3 - f);
31
32
        % Plotting
33
        figure;
34
35
        subplot(3,1,1);
36
        plot(t, error1);
37
        title('Absolute Error for T = 0.5s');
38
        xlabel('t');
39
        xlim([-3.5, 3.5]);
        ylabel('Absolute Error');
40
41
        ylim([0, 0.015]);
42
        grid on;
43
        subplot(3,1,2);
44
45
        plot(t, error2);
46
        title('Absolute Error for T = 1.0s');
47
        xlabel('t');
48
        x1im([-3.5, 3.5]);
49
        ylabel('Absolute Error');
50
        ylim([0, 0.015]);
51
        grid on;
52
53
        subplot(3,1,3);
54
        plot(t, error3);
        title('Absolute Error for T = 2.0s');
55
56
        xlabel('t');
        xlim([-3.5, 3.5]);
57
```

```
58    ylabel('Absolute Error');
59    ylim([0, 0.35]);
60    grid on;
61    end
62
```

2.2 对实验1中的信号,绘制抽样间隔为1s下的抽样信号经过DAC一阶保持器后的恢复信号时域波形,体会一阶保持器的基本原理和作用。



```
1
   function ex3_2_2()
2
       % 抽样间隔为 1s
3
       T = 1;
4
       n = -pi:pi;
       t = n * T;
5
6
       f = 0.5 * (1 + cos(t));
7
8
       % 绘制抽样信号和恢复信号的时域波形
9
       figure:
10
       plot(t, f, 'Linewidth', 2, 'DisplayName', '抽样信号');
11
       stem(t, f, 'Marker', 'o', 'LineWidth', 1.5, 'Color', 'r', 'DisplayName',
12
    '恢复信号');
       hold off;
13
14
       %添加标题和标签
15
16
       title('抽样信号经过 DAC 一阶保持器后的恢复信号时域波形');
17
       xlabel('时间 (t)');
18
       xlim([-3.5, 3.5]);
19
       ylabel('幅度 (f(t))');
20
       % 打开网格和添加图例
21
22
       grid on;
23
       legend('Location', 'Best');
24
    end
```

五、实验总结

在验证抽样定理的过程中,抽样间隔的选择对信号混叠的影响十分关键。当抽样间隔为0.5s时,大于奈奎斯特抽样频率,相邻信号之间没有发生混叠。而当抽样间隔为2s时,小于奈奎斯特抽样频率,相邻信号之间发生混叠,导致信号失真。这验证了抽样定理的正确性。

另外,对信号的恢复重建过程也进行了探讨。选择合适的抽样间隔是保证信号恢复效果的关键。较小的抽样间隔(对应较大的抽样频率)通常会导致更好的信号恢复效果,而较大的抽样间隔则可能导致信号失真。因此,在实践中需要平衡采样频率和信号失真的需求,以确保在可容忍范围内取得最佳效果。

通过这个实验,不仅巩固了信号的抽样定理和恢复原理,还通过 Matlab 编程锻炼了实际应用的能力。这样的实践有助于深入理解理论知识并提高解决实际问题的能力。

教师评语:	
签名:	
日期:	

成绩: