

西安电子科技大学

数字信号处理 实验报告

实验名称 信号的采样与重建

人工智能 学院 23200XX 班

姓名 X X X 学号 2300920XXX

X X X 2300920XXX

X X X 2300920XXX

实验报告内容基本要求及参考格式

一、实验目的

二、实验所用仪器（或实验环境）

三、实验基本原理及步骤

四、实验结果

五、实验代码

六、实验问题及解决方案

张三：

李四：

王五：

一、实验目的

1. 掌握连续时间模拟信号的数学建模与计算机仿真产生方法;
2. 理解均匀采样过程、过采样与欠采样的区别;
3. 通过波形对比深入理解奈奎斯特-香农采样定理;
4. 掌握理想低通滤波器重建（sinc 内插）原理并实现信号恢复;
5. 比较不同采样率下重建信号的质量。

二、实验所用仪器（或实验环境）

1. 需要环境为： Python 3.x
2. 需要安装的 Python 库： numpy, matplotlib

三、实验基本原理及步骤

1. 实验基本原理

- 1) 模拟信号：选取含多个频率分量的信号

$$x(t) = \cos(2\pi \times 50t) + 0.5 \times \cos(2\pi \times 120t) + 0.3 \times \cos(2\pi \times 300t)$$

- 2) 最高频率 $f_{max} = 300$ Hz，按照采样定理，理论最小采样频率 $f_s \geq 2 \times 300 = 600$ Hz。

- 3) 采样过程： $x[n] = x(nT_S)$, $T_S = \frac{1}{f_s}$

- 4) 理想重建公式（sinc 内插）： $\hat{x}(t) = \sum x[n] \cdot \text{sinc}(fs(t - nT_s))$

2. 实验步骤

- 1) 产生模拟信号（高密度时间轴用于显示“连续”波形）

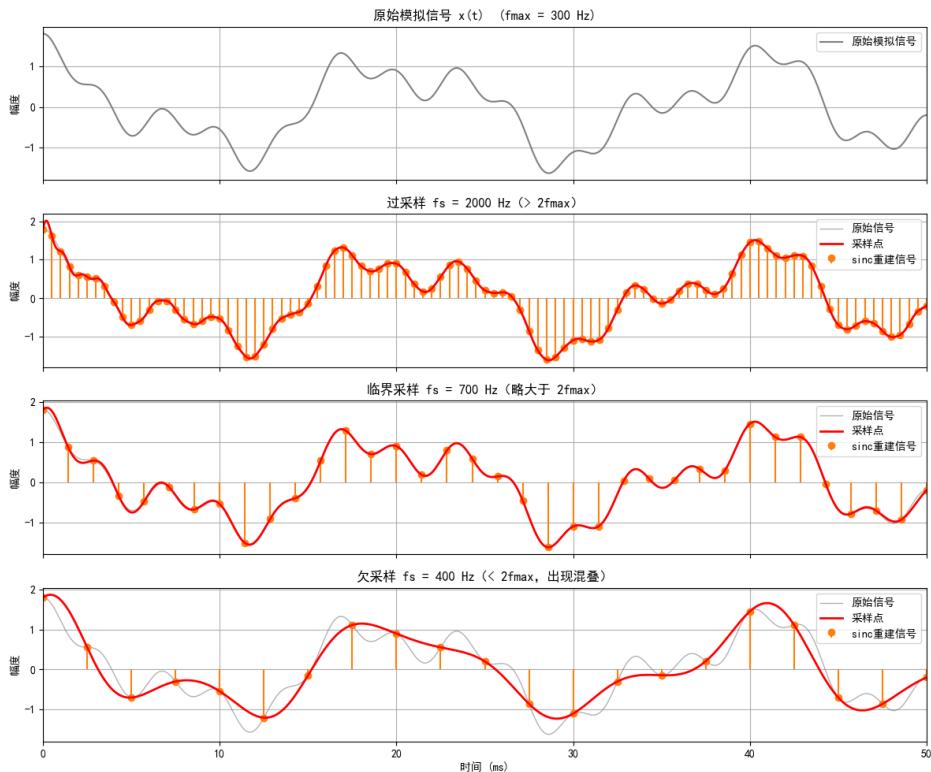
- 2) 选取三种采样频率

- 过采样： $fs_1 = 2000$ Hz ($> 2f_{max}$)
- 临界采样： $fs_2 = 700$ Hz (略大于 $2f_{max}$)
- 欠采样： $fs_3 = 400$ Hz ($< 2f_{max}$, 会产生明显混叠)

3) 对三种采样信号分别进行 sinc 重建

4) 绘制对比波形图

四、实验结果



实验结果说明:

1. $f_s = 2000$ Hz (过采样)

采样点密集, sinc 重建信号与原始信号几乎完全重合, 无失真。

2. $f_s = 700$ Hz (略大于 $2f_{\max}$)

采样点较稀疏, 重建信号基本恢复原始波形, 但高频分量 300Hz 处有轻微吉布斯现象 (波纹)。

3. $f_s = 400$ Hz (欠采样)

出现严重频谱混叠 (300Hz 分量折叠到 100Hz 等), 重建信号完全失真, 与原始信号波形差异巨大。

结果分析

1. 当 $fs \geq 2f_{max}$ 时，可以无失真地从采样信号恢复原始模拟信号（带限信号前提下）；
2. 当 $fs < 2f_{max}$ 时，必然产生混叠失真，且混叠后无法通过任何线性滤波器分离；
3. 实际工程中通常采用 $fs = (4\sim 10) \times f_{max}$ 的过采样以留有余量并便于抗混叠滤波器设计；
4. sinc 函数重建是理论上的理想重建，实际中用有限阶低通滤波器近似，会引入一定误差。

五、实验代码

程序代码：

```

import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei', 'Arial Unicode MS', 'DejaVu Sans']
plt.rcParams['axes.unicode_minus'] = False

# 参数设置
f1, f2, f3 = 50, 120, 300          # 三分量频率 (Hz)
A1, A2, A3 = 1.0, 0.5, 0.3
T_total = 0.05                      # 显示 50 ms

# 高密度时间轴，用于绘制“连续”模拟信号
t_fine = np.linspace(0, T_total, 100000)
x_analog = A1*np.cos(2*np.pi*f1*t_fine) + \
            A2*np.cos(2*np.pi*f2*t_fine) + \
            A3*np.cos(2*np.pi*f3*t_fine)

# 三种采样频率
fs_list = [2000, 700, 400]
labels = ['过采样 fs = 2000 Hz (> 2fmax)', 
          '临界采样 fs = 700 Hz (略大于 2fmax)', 
          '欠采样 fs = 400 Hz (< 2fmax, 出现混叠)']

```

```

# 创建 4 行 1 列的子图

fig, axs = plt.subplots(4, 1, figsize=(12, 10), sharex=True)

# 第一幅图：原始模拟信号

axs[0].plot(t_fine*1000, x_analog, 'gray', lw=1.5)
axs[0].set_title('原始模拟信号 x(t) (fmax = 300 Hz)')
axs[0].set_ylabel('幅度')
axs[0].grid(True)
axs[0].legend(['原始模拟信号'], loc='upper right')

for i, fs in enumerate(fs_list):

    Ts = 1 / fs

    n = np.arange(0, int(T_total * fs + 1)) # 采样点序号

    t_sample = n * Ts

    x_sample = A1*np.cos(2*np.pi*f1*t_sample) + \
               A2*np.cos(2*np.pi*f2*t_sample) + \
               A3*np.cos(2*np.pi*f3*t_sample)

    # sinc 重建（理想低通滤波器重建）

    t_recon = np.linspace(0, T_total, 10000)
    x_recon = np.zeros_like(t_recon)

    for k in range(len(t_recon)):

        x_recon[k] = np.sum(x_sample * np.sinc(fs * (t_recon[k] - t_sample)))

    # 绘图

    ax = axs[i+1]

    # 灰色细线：原始信号（参考）

    ax.plot(t_fine*1000, x_analog, color='gray', lw=1, alpha=0.6)

    # 蓝色茎图：采样点

    ax.stem(t_sample*1000, x_sample, linefmt='C1-', markerfmt='C1o', basefmt="")

    # 红色实线：sinc 重建信号

    ax.plot(t_recon*1000, x_recon, 'r-', lw=2)

```

```

ax.set_title(labels[i])
ax.set_ylabel('幅度')
ax.grid(True)
ax.legend(['原始信号', '采样点', 'sinc 重建信号'])
ax.set_xlim(0, T_total*1000)
axs[3].set_xlabel('时间 (ms)')
plt.tight_layout()
plt.savefig('信号采样与 sinc 重建对比图.png', dpi=300)
plt.show()

```

六、实验问题及解决方案

	实验问题	解决方案
XXX	建立模拟信号的数学模型，设计计算机程序仿真产生模拟信号	用 Python 的 numpy 库构造多频率余弦信号叠加的数学模型；生成高密度时间轴，计算模拟信号的时域波形
XXX	采用过采样和欠采样多个不同的采样频率对模拟信号进行时域采样产生离散信号	设定过采样（如 2000Hz）、临界采样（如 700Hz）、欠采样（如 400Hz）频率；按采样频率生成采样时刻，计算对应时刻的信号值得到离散采样点
XXX	绘制模拟信号和离散信号的时域波形图进行分析对比，并对采样信号进行恢复，绘制内插后的模拟信号	用 matplotlib 绘制原始模拟信号、采样点（茎图）；用 sinc 内插法（理想低通重建）恢复采样信号，绘制重建后的波形并与原始信号对比