



课程项目

危国锐 516021910080

(上海交通大学电子信息与电气工程学院, 上海 200240)

摘要: 本文通过推导四种 Fourier 变换, 说明: (1) 非周期信号的 Fourier 变换可视为相应周期信号情形的推广, (2) 周期 (非周期)、离散 (连续) 时间信号的频谱是离散 (连续)、周期 (非周期) 的. 本文从数值积分的视角, 观察离散 Fourier 变换 (DFT) 和连续时间 Fourier 变换 (CTFT) 的定义式, 发现 DFT 可被视作时域有限信号的 CTFT 的一种离散方式, 据此建立了一种用 DFT 数值计算时域有限信号的数值 CTFT 的算法. 推导了数值 CTFT 的反变换性质, 由此建立了一种用离散 Fourier 逆变换 (IDFT) 数值计算连续时间 Fourier 逆变换 (ICTFT) 的算法, 与用 DFT 计算数值 CTFT 的算法互逆. 推导了数值 CTFT 的尺度变换性质, 时移性质和奇偶虚实性, 发现它们与解析 CTFT 的相应性质一致. 本文从理论上分析了经典的理想冲激串采样-重建流程, 由此提出了一种基于理想带限内插的信号重建方法. 将这方法应用于一个简单的连续时间周期信号, 发现: (1) 即使采样频率高于原始信号对应的 Nyquist 频率, 也往往不能通过该信号在有限区间上的等间隔采样值精确重建该区间内的原始信号, (2) 根据时域无限信号在有限采样时间区间上的等间隔样值, 用前述信号重建方法在该区间内重建原始信号, 隐含着信号在采样区间边缘存在一个从非零到零的突变, 这种突变对应着频率约为采样频率的一半的频率分量, 这是重建误差的来源之一.

关键词: Fourier 变换, 频域分析, 采样



Course Project

Guorui Wei 516021910080

(School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: In this paper, by deriving four Fourier transforms, we show that (1) the Fourier transform of a nonperiodic signal can be regarded as a generalization of the corresponding periodic signal case, and (2) the spectrum of periodic (nonperiodic) and discrete (continuous) time signals is discrete (continuous) and periodic (nonperiodic). In this paper, from the viewpoint of numerical integration, we observe the defining equations of discrete Fourier transform (DFT) and continuous time Fourier transform (CTFT), and find that DFT can be regarded as a discrete way of CTFT for finite signals in the time domain, whereby an algorithm for numerical computation of numerical CTFT for finite signals in the time domain by DFT is established. The inverse transform property of numerical CTFT is derived, and an algorithm for numerical computation of continuous-time Fourier inverse transform (ICTFT) with discrete Fourier inverse transform (IDFT) is thus established, which is reciprocal to the algorithm for numerical computation of numerical CTFT with DFT. The scale-transform properties, time-shift properties and parity falsity of numerical CTFT are derived and found to be consistent with the corresponding properties of analytic CTFT. In this paper, the classical ideal impulse string sampling-reconstruction process is analyzed theoretically, and a signal reconstruction method based on ideal band-limited interpolation is proposed. Applying this method to a simple continuous time period signal, it is found that (1) even if the sampling frequency is higher than the Nyquist frequency corresponding to the original signal, the original signal within the interval cannot often be accurately reconstructed from the equally spaced sample values of this signal over a finite interval, and (2) based on the equally spaced sample values of an infinite signal in the time domain over a finite sampling time interval, the reconstruction of the original signal with the aforementioned signal reconstruction method within the interval The reconstruction of the original signal within the interval implicitly implies the existence of a sudden change of the signal from non-zero to zero at the edge of the sampling interval, and this sudden change corresponds to a frequency component whose frequency is about half of the sampling frequency, which is one of the sources of the reconstruction error.

Keywords: Fourier transform, frequency-domain analysis, sampling



目 录

摘要	i
Abstract.....	ii
目 录	iii
1 项目要求	1
2 预备知识	2
2.1 离散 Fourier 变换 (DFT)	2
2.2 离散时间 Fourier 变换 (DTFT)	2
2.3 Fourier 级数 (FS)	3
2.4 连续时间 Fourier 变换 (CTFT)	3
2.5 DFT 与 CTFT 的关系: 从数值积分的视角.....	4
3 项目 1	6
3.1 项目简介.....	6
3.2 数据和方法.....	6
3.2.1 音频信号.....	6
3.2.2 CTFT 数值算法.....	6
3.2.3 数值 CTFT 的性质	6
3.3 结果和讨论.....	9
3.4 项目小结.....	15
4 项目 2: 语音采样器 (Speech Sampler)	16
4.1 项目简介.....	16
4.2 数据和方法.....	16
4.2.1 连续时间信号.....	16
4.2.2 冲激串采样与理想带限内插.....	17
4.2.3 基于理想带限内插的信号重建.....	19
4.2.4 信号重建质量的评价.....	19
4.3 结果和讨论.....	20
4.3.1 重建的信号.....	20
4.3.2 讨论.....	20
4.4 项目小结.....	24
References	25
Appendix A MATLAB 源代码	26
A.1 proj1.m	26
A.2 proj2_2.m	37



1 项目要求

目标:

- 帮助学生理解和掌握课程的基本理论
- 提升学生将理论应用于实践中的能力

软件需求:

工具: MATLAB (推荐), 或其它工具, 例如 Python, C/C++

作业要求:

- 请在 CANVAS 上提交一个.zip 格式的文件, 你的解答应至少包含以下三方面的内容:
 - 1) 一份大作业报告 (中文或英文均可, .pdf 格式);
 - 2) 结果中的语音片段;
 - 3) 用于生成上述语音片段的源代码.
- 不允许迟交, 否则会酌情扣分

项目 1 (5 分)

- 1) 选择一段音频信号 (音乐等), 将其记为 $f(t)$, 画出该信号的波形.
- 2) 生成 $f(-t), f(2t), f(t/2)$, 并画出它们的波形.
- 3) 分别计算 $f(t), f(2t), f(t/2)$ 的傅里叶变换, 画出它们的频谱, 并对它们进行比较和分析.
- 4) 将 $f(t)$ 的傅里叶变换记为 $F(j\omega)$, 画出幅度谱的傅里叶反变换 $\mathcal{F}^{-1}\{|F(j\omega)|\}$ 的波形, 画出相位谱的傅里叶反变换 $\mathcal{F}^{-1}\{e^{j\angle F(j\omega)}\}$ 的波形. 将它们与原始信号 $f(t)$ 进行比较.
- 5) 对 $f(t)$ 在频域实现一个低通滤波器 (可使用理想低通滤波器, 自行确定截止频率), 画出得到的信号的波形.

项目 2 (5 分, 以下两个选项中任选一个)

提示: 你可以自由选取所用的声音信号.

选项 1: 声音消除器 (Voice Eliminator)

- 1) 设计一个名为“Voice Eliminator”的软件或仿真程序, 可用于消除一首歌曲中歌手的声音.
- 2) 分析“Voice Eliminator”的基本设计思路和原理, 并用编程语言实现之.
- 3) 进一步, 使用合适的方法来提高“Voice Eliminator”的性能 (从理论和实际的角度).

选项 2: 语音采样器 (Speech Sampler)

- 1) 采集某个人的声音作为一段连续时间信号, 使用不同的采样频率对其进行若干次采样, 得到若干份离散时间信号. (推荐采样频率: 44 kHz, 22 kHz, 11 kHz, 5.5 kHz, 2.75 Hz)
- 2) 对上述离散时间信号分别进行重构, 得到相应的连续时间信号, 分析不同采样频率对重构质量的影响, 并计算重构误差.
- 3) 分析“Speech Sampler”的基本设计思路和原理, 并用编程语言实现之.



2 预备知识

本节首先推导四种 Fourier 变换, 得到: 1) 非周期信号的 Fourier 变换可视作周期信号情形的推广, 2) 周期 (非周期)、离散 (连续) 时间信号的频谱是离散 (连续)、周期 (非周期) 的. 然后, 从数值积分的角度, 说明对于一个连续时间信号, 其采样值序列的离散 Fourier 变换可视作该信号的 Fourier 变换的一个近似 (2.24).

2.1 离散 Fourier 变换 (DFT)

在一定的条件下, 以正整数 N 为最小正周期的离散时间信号 $x(n)$ 可表为 N 个以 N 为周期的复指数信号

$$\exp\left(jk\frac{2\pi}{N}n\right), \quad k=0,1,\dots,N-1$$

的线性组合:

$$x(n) = \sum_{k=0}^{N-1} a_k \exp\left(jk\frac{2\pi}{N}n\right). \quad (2.1)$$

若将 a_k 视作 $x(n)$ 在函数内积空间

$$\text{span}\left\{\exp\left(jk\frac{2\pi}{N}n\right)\right\}, \quad k=0,1,\dots,N-1$$

上的最佳平方逼近的坐标, 则 a_k 满足法方程

$$\Phi \mathbf{a} = \mathbf{b}, \quad (2.2)$$

其中 $\mathbf{a} := (a_0, \dots, a_{N-1})^T$, $\mathbf{b} := (b_0, \dots, b_{N-1})^T$, $\Phi := (\varphi_{ij})_{N \times N}$,

$$b_k := \left\langle x(n), \exp\left(jk\frac{2\pi}{N}n\right) \right\rangle := \sum_{n=0}^{N-1} x(n) \exp\left(-jk\frac{2\pi}{N}n\right), \quad (2.3)$$

$$\varphi_{ij} := \left\langle \exp\left(ji\frac{2\pi}{N}n\right), \exp\left(jj\frac{2\pi}{N}n\right) \right\rangle := \sum_{n=0}^{N-1} \exp\left(ji\frac{2\pi}{N}n\right) \exp\left(-jj\frac{2\pi}{N}n\right) = N\delta_{ij}. \quad (2.4)$$

从而

$$a_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \exp\left(-jk\frac{2\pi}{N}n\right). \quad (2.5)$$

定义

$$\mathcal{F}[x(n)](k) := \hat{x}(k) := \sum_{n=0}^{N-1} x(n) \exp\left(-jk\frac{2\pi}{N}n\right), \quad k=0,1,\dots,N-1 \quad (2.6)$$

为离散时间序列 $x(n)$ 的 N 点离散 Fourier 变换 (Discrete Fourier transform, DFT), 则

$$\mathcal{F}^{-1}[\hat{x}(k)](n) := x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}(k) \exp\left(jk\frac{2\pi}{N}n\right) \quad (2.7)$$

为相应的 N 点离散 Fourier 逆变换 (Inverse discrete Fourier transform, IDFT).

可见, 周期、离散时间信号的频谱 (Fourier 变换) 是离散的、周期的.

2.2 离散时间 Fourier 变换 (DTFT)

将周期的离散时间序列的 DFT 推广到非周期的离散时间序列, 即得离散时间 Fourier 变换 (DTFT).

设 $x(n)$ 是一个非周期的离散时间信号. 定义以偶数 N 为最小正周期的离散时间信号



$$x_N(n) := \begin{cases} x(n), & -\frac{N}{2} \leq n < \frac{N}{2}, \\ x(n+N), & \forall n. \end{cases} \quad (2.8)$$

有

$$x(n) = \lim_{N \rightarrow +\infty} x_N(n), \quad \forall n. \quad (2.9)$$

由 IDFT (2.7), 有

$$\begin{aligned} x(n) &= \lim_{N \rightarrow +\infty} x_N(n) \\ &= \lim_{N \rightarrow +\infty} \frac{1}{N} \sum_{k=-N/2}^{N/2-1} \exp\left(jk \frac{2\pi}{N} n\right) \sum_{n=-N/2}^{N/2-1} x(n) \exp\left(-jk \frac{2\pi}{N} n\right) \\ &= \frac{1}{2\pi} \int_{2\pi} \exp(j\omega n) d\omega \sum_{n=-\infty}^{+\infty} x(n) \exp(-j\omega n). \end{aligned} \quad (2.10)$$

定义

$$\mathcal{F}[x(n)](\omega) := \hat{x}(\omega) := \sum_{n=-\infty}^{+\infty} x(n) \exp(-j\omega n), \quad \omega \in \mathbb{R} \quad (2.11)$$

为离散时间序列 $x(n)$ 的**离散时间 Fourier 变换** (Discrete-time Fourier transform, DTFT), 则

$$\mathcal{F}^{-1}[\hat{x}(\omega)](n) := x(n) = \frac{1}{2\pi} \int_{2\pi} \hat{x}(\omega) \exp(j\omega n) d\omega \quad (2.12)$$

为相应的**离散时间 Fourier 逆变换**. 这样, 非周期的离散时间信号 $x(n)$ 被表为复指数信号 $\exp(j\omega n)$ 的“线性组合”.

注意到 2π 是 $\mathcal{F}[x(n)](\omega)$ 的周期. 可见, 非周期、离散时间信号的频谱是连续的、周期的.

2.3 Fourier 级数 (FS)

在一定的条件下, 以 T 为最小正周期的连续时间信号 $x(t)$ 可表为可列个以 T 为周期的复指数信号

$$\exp(jk\omega_0 t), k \in \mathbb{Z}, \quad \omega_0 := 2\pi/T$$

的线性组合:

$$x(t) = \sum_{k \in \mathbb{Z}} \exp(jk\omega_0 t) \frac{1}{T} \int_T x(t) \exp(-jk\omega_0 t) dt. \quad (2.13)$$

称 (2.13) 为周期的连续时间信号 $x(t)$ 的**Fourier 级数** (Fourier series, FS).

可见, 周期、连续时间信号的频谱是离散的、非周期的.

2.4 连续时间 Fourier 变换 (CTFT)

将周期的连续时间信号的 Fourier 级数推广到非周期的连续时间信号, 即得连续时间 Fourier 变换 (CTFT).

设 $x(t)$ 是一个非周期的连续时间信号. 定义以 T 为最小正周期的连续时间信号

$$x_T(t) := \begin{cases} x(t), & -\frac{T}{2} \leq t < \frac{T}{2}, \\ x(t+T), & \forall t. \end{cases} \quad (2.14)$$

有



$$x(t) = \lim_{T \rightarrow +\infty} x_T(t), \quad \forall t. \quad (2.15)$$

由周期的连续时间信号的 Fourier 级数 (2.13), 有

$$\begin{aligned} x(t) &= \lim_{T \rightarrow +\infty} x_T(t) \\ &= \lim_{T \rightarrow +\infty} \frac{1}{T} \sum_{k \in \mathbb{Z}} \exp(jk\omega_0 t) \int_{-T/2}^{T/2} x(t) \exp(-jk\omega_0 t) dt \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} \exp(j\omega t) d\omega \int_{\mathbb{R}} x(t) \exp(-j\omega t) dt. \end{aligned} \quad (2.16)$$

定义

$$\mathcal{F}[x(t)](\omega) := \hat{x}(\omega) := \int_{\mathbb{R}} x(t) \exp(-j\omega t) dt, \quad \omega \in \mathbb{R} \quad (2.17)$$

为连续时间信号 $x(t)$ 的**连续时间 Fourier 变换** (Continuous-time Fourier transform, CTFT), 则

$$\mathcal{F}^{-1}[\hat{x}(\omega)](t) := x(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{x}(\omega) \exp(j\omega t) d\omega \quad (2.18)$$

为相应的**连续时间 Fourier 逆变换** (Inverse continuous-time Fourier transform, ICTFT). 这样, 非周期的连续时间信号 $x(t)$ 被表为复指数信号 $\exp(j\omega t)$ 的“线性组合”.

可见, 非周期、连续时间信号的频谱是连续的、非周期的.

2.5 DFT 与 CTFT 的关系: 从数值积分的视角

自然界中的声音信号是连续的, 但计算机通常只能以离散方式存储和处理之.

设有连续时间信号 $x(t)$ 满足

$$x(t) = 0, \quad \forall t: t < t_0 \text{ or } t \geq t_0 + \tau. \quad (2.19)$$

这 $x(t)$ 仅在有限时间上不恒为零, 故称其为**时域有限的**. 记 $x(t)$ 的 CTFT 为

$$X(\omega) := \mathcal{F}[x(t)] = \int_{\mathbb{R}} x(t) \exp(-j\omega t) dt = \int_{t_0}^{t_0+\tau} x(t) \exp(-j\omega t) dt. \quad (2.20)$$

设在计算机中存储了 $x(t)$ 在 $t = t_0 + nT_s$, $n = 0, \dots, N-1$, 时刻的采样值 $x(t_0 + nT_s)$, $NT_s = \tau$. 记 $F_s := 1/T_s$ 为采样频率, 则 $N = \tau F_s$. 记离散时间序列

$$\tilde{x}(n) := x(t_0 + nT_s), \quad 0 \leq n < N \quad (2.21)$$

的 N 点 DFT (2.6) 为

$$\hat{x}(k) := \mathcal{F}[\tilde{x}(n)] = \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp\left(-jk \frac{2\pi}{N} n\right), \quad k = 0, 1, \dots, N-1. \quad (2.22)$$

使用采样时刻 $t = t_0 + nT_s$, $n = 0, \dots, N-1$, 上 $x(t)$ 的值, 定积分 (2.20) 的矩形近似为

$$X(\omega) \sim T_s \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp(-j\omega(t_0 + nT_s)). \quad (2.23)$$

比较 (2.22) 和 (2.23), 得 $x(t)$ 的 CTFT $X(\omega)$ 与其采样序列 $\tilde{x}(n) := x(nT_s)$ 的 N 点 DFT $\hat{x}(k)$ 的关系为 (在矩形积分公式近似下)

$$X(\omega)|_{\omega=\frac{k}{N}\omega_s} \sim T_s \hat{x}(k) \exp\left(-j \frac{k}{N} \omega_s t_0\right), \quad k = 0, 1, \dots, N-1, \quad (2.24)$$

其中 $\omega_s := 2\pi F_s$ 为采样圆频率. 记 $\omega_k := \frac{k}{N} \omega_s$, 上式写成



$$X(\omega_k) \sim T_s \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp(-j\omega_k(t_0 + nT_s)). \quad (2.24')$$

式(2.24)右端可作为 $x(t)$ 的 CTFT $X(\omega)$ 的一个数值解, 即 DFT 可被视作时域有限信号的 CTFT 的一种离散方式.

注 2.1 由(2.24)只能得到 $X(\omega)$ 在 $\omega \in [0, \omega_s)$ 上的一些近似值(若取 $-N/2 \leq k < N/2$, 则得到 $\omega \in [-\omega_s/2, \omega_s/2)$ 上的), 这符合 Nyquist 采样定理. ■

注 2.2 事实上, 不必限制(2.22)和(2.24)中 $k \in \mathbb{Z}$ 的取值. 因为 $\hat{x}(k)$ 具有周期 N , 所以 CTFT 数值解 $X(\omega)|_{\omega=\frac{k}{N}\omega_s}$ 具有周期 ω_s . ■



3 项目 1

3.1 项目简介

选择一段音频信号 (音乐等), 将其记为 $f(t)$, 画出该信号的波形.

生成 $f(-t), f(2t), f(t/2)$, 并画出它们的波形.

分别计算 $f(t), f(2t), f(t/2)$ 的傅里叶变换, 画出它们的频谱, 并对它们进行比较和分析.

将 $f(t)$ 的傅里叶变换记为 $F(j\omega)$, 画出幅度谱的傅里叶反变换 $\mathcal{F}^{-1}\{|F(j\omega)|\}$ 的波形, 画出相位谱的傅里叶反变换 $\mathcal{F}^{-1}\{e^{j\angle F(j\omega)}\}$ 的波形. 将它们与原始信号 $f(t)$ 进行比较.

对 $f(t)$ 在频域实现一个低通滤波器 (可使用理想低通滤波器, 自行确定截止频率), 画出得到的信号的波形.

3.2 数据和方法

3.2.1 音频信号

本项目使用的音频信号是使用 [Chrome Music Lab - Song Maker \(chromeexperiments.com\)](https://musiclab.chromeexperiments.com/Song-Maker/song/6679651116384256) 工具¹制作的, 时长 2 秒, 采样率 48 kHz, 可从 <https://musiclab.chromeexperiments.com/Song-Maker/song/6679651116384256> 下载.

3.2.2 CTFT 数值算法

设在计算机中存储了连续时间信号 $x(t)$ 在 $t = t_0 + nT_s, n = 0, \dots, N-1$ 时刻的样值

$$x(t_0 + nT_s), \quad n = 0, \dots, N-1,$$

采样间隔 $T_s > 0$. 下面建立数值求解 $x(t)$ 的 CTFT 的算法.

假定 $x \in C_c[t_0, t_0 + NT_s]$ ². $x(t)$ 的 CTFT 可按 (2.24) 求得数值结果:

$$X(\omega)|_{\omega=\frac{k}{N}\omega_s} \sim \exp\left(-j\frac{k}{N}\omega_s t_0\right) T_s \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp\left(-jk\frac{2\pi}{N}n\right) \quad (3.1)$$

上式右端的求和号可直接调用快速 Fourier 变换程序 (例如, Matlab 的 `fft` 函数³) 计算.

至此, 数值求解 $x(t)$ 的 CTFT 的算法已建立, 现重述如下.

算法 3.1 (CTFT by DFT) 输入: 数组 $\{t_n\}, \{x_n\}, n = 0, \dots, N-1$. 要求 $\{t_n\}$ 单调递增, 且等间隔.

步 1 决定 t_0, T_s . 令 t_0 为数组 $\{t_n\}$ 的首元素, T_s 为 $\{t_n\}$ 的公差.

步 2 按 (3.1) 计算 $X(\omega)|_{\omega=\frac{k}{N}\omega_s} \sim \exp(-j\frac{k}{N}\omega_s t_0) T_s \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp(-jk\frac{2\pi}{N}n)$.

■

3.2.3 数值 CTFT 的性质

记 $X(\omega)$ 是用算法 3.1 求出的 $x(t)$ 的数值 CTFT, 下面推导其反变换, 时移, 尺度变换和奇偶虚实性质, 并与 CTFT 解析解的相应性质作比较.

¹ <https://musiclab.chromeexperiments.com/Song-Maker/>

² continuous functions that are only nonzero on a closed subinterval of $[t_0, t_0 + NT_s)$

³ <https://ww2.mathworks.cn/help/matlab/ref/fft.html>



设 $X(\omega)$ 是由 $x(t)$ 的样值

$$x(t_0 + nT_s), \quad n = 0, \dots, N-1$$

用算法 3.1 求得的, 简记为

$$x(t) \xrightleftharpoons[\text{算法 3.1}]{\text{数值 CTFT}} X(\omega_k),$$

其中 $\omega_k := \frac{k}{N}\omega_s$, $k \in \mathbb{Z}$, $\omega_s := 2\pi/T_s$,

$$X(\omega_k) := \exp(-j\omega_k t_0) T_s \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp(-j\omega_k nT_s). \quad (3.1')$$

1 反变换

由离散 Fourier 逆变换 (2.7), 有

$$\begin{aligned} x(t_0 + nT_s) &= \frac{1}{N} \sum_{k=0}^{N-1} \exp(j\omega_k nT_s) \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp(-j\omega_k nT_s) \\ &= \frac{1}{2\pi} \frac{\omega_s}{N} \sum_{k=0}^{N-1} X(\omega_k) \exp(j\omega_k(t_0 + nT_s)). \end{aligned} \quad (3.2)$$

上式右端可视作连续时间 Fourier 逆变换 (2.18)

$$x(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{x}(\omega) \exp(j\omega t) d\omega \quad (3.3)$$

的矩形积分公式的近似, 其中 $\hat{x}(\omega)$ 是 $x(t)$ 的 CTFT 解析解 (2.17)

$$\hat{x}(\omega) := \mathcal{F}[x(t)] := \int_{\mathbb{R}} x(t) \exp(j\omega t) dt. \quad (2.17)$$

式 (3.2) 可写成

$$x(t_0 + nT_s) = F_s \frac{1}{N} \sum_{k=0}^{N-1} (X(\omega_k) \exp(j\omega_k t_0)) \exp\left(jk \frac{2\pi}{N} n\right). \quad (3.2')$$

上式右端的求和号可直接调用快速 Fourier 逆变换程序 (例如, Matlab 的 ifft 函数⁴) 计算.

算法 3.2 (ICTFT by IDFT) 输入: 初始时刻 t_0 , 数组 $\{\omega_k\}, \{X_k\}$, $k = 0, \dots, N-1$. 要求 $\{\omega_k\}$ 单调递增, 且等间隔.

步 1 决定 F_s, T_s . 令 ω_s 为 $\{\omega_k\}$ 的公差, 则 $F_s := \omega_s/(2\pi)$, $T_s := 2\pi/\omega_s$.

步 2 按 (3.2') 计算 $x(t_0 + nT_s) = F_s \frac{1}{N} \sum_{k=0}^{N-1} (X(\omega_k) \exp(j\omega_k t_0)) \exp(jk \frac{2\pi}{N} n)$. ■

2 尺度变换性质

考虑信号 $g(t) := x(at)$. 若 $a > 0$, 则它的数值 CTFT $G(\omega)$ 将由其样值

$$g\left(\frac{t_0}{a} + n \frac{T_s}{a}\right) = x(t_0 + nT_s), \quad n = 0, \dots, N-1$$

决定. 根据算法 3.1,

$$\begin{aligned} G(\omega)|_{\omega=\frac{k}{N}a\omega_s} &= \exp\left(-j \frac{k}{N} a \omega_s \frac{t_0}{a}\right) \frac{T_s}{a} \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp\left(-jk \frac{2\pi}{N} n\right) \\ &= \frac{1}{a} X(\omega)|_{\omega=\frac{k}{N}\omega_s}, \quad a > 0, k \in \mathbb{Z}. \end{aligned} \quad (3.4)$$

若 $a < 0$, 则它的 CTFT 数值解 $G(\omega)$ 将由其样值

$$g\left(\frac{t_0}{a} + (N-1) \frac{T_s}{a} + n \frac{T_s}{-a}\right) = x(t_0 + (N-1-n)T_s), \quad n = 0, \dots, N-1$$

⁴ <https://ww2.mathworks.cn/help/matlab/ref/iff.html>



决定. 根据算法 3.1, 此时

$$\begin{aligned}
 G(\omega)|_{\omega=\frac{k}{N}(-a)\omega_s} &= \exp\left(-j\frac{k}{N}(-a)\omega_s\left(\frac{t_0}{a}+(N-1)\frac{T_s}{a}\right)\right) \\
 &\times \frac{T_s}{-a} \sum_{n=0}^{N-1} x(t_0+(N-1-n)T_s) \exp\left(-jk\frac{2\pi}{N}n\right) \\
 &= \exp\left(-j\frac{-k}{N}\omega_s t_0\right) \exp\left(-j\frac{k}{N}\omega_s(-(N-1)T_s)\right) \\
 &\times \frac{T_s}{-a} \sum_{n=0}^{N-1} x(t_0+nT_s) \exp\left(-jk\frac{2\pi}{N}(N-1-n)\right) \\
 &= \exp\left(-j\frac{-k}{N}\omega_s t_0\right) \frac{T_s}{-a} \sum_{n=0}^{N-1} x(t_0+nT_s) \exp\left(-j(-k)\frac{2\pi}{N}n\right) \\
 &= \frac{1}{-a} X(\omega)|_{\omega=\frac{-k}{N}\omega_s}, \quad a < 0, k \in \mathbb{Z}.
 \end{aligned} \tag{3.5}$$

式 (3.4) (3.5) 可统一写成

$$g(t) := x(at) \xleftrightarrow[\text{算法 3.1}]{\text{数值 CTFT}} G(\omega_k) = \frac{1}{|a|} X\left(\frac{\omega_k}{a}\right), \quad a \neq 0, \tag{3.6}$$

其中 $\omega_k := \frac{k}{N}|a|\omega_s$, $k \in \mathbb{Z}$.

式 (3.6) 与 CTFT 解析解的尺度变换性质相同.

3 时移性质

考虑信号 $g(t) := x(t + \tau)$, 则它的数值 CTFT 数值 $G(\omega)$ 将由其样值

$$g(t_0 - \tau + nT_s) = x(t_0 + nT_s), \quad n = 0, \dots, N-1$$

决定. 根据算法 3.1,

$$\begin{aligned}
 G(\omega)|_{\omega=\frac{k}{N}\omega_s} &= \exp\left(-j\frac{k}{N}\omega_s(t_0 - \tau)\right) T_s \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp\left(-jk\frac{2\pi}{N}n\right) \\
 &= \exp\left(j\frac{k}{N}\omega_s\tau\right) X(\omega)|_{\omega=\frac{k}{N}\omega_s}, \quad k \in \mathbb{Z}.
 \end{aligned} \tag{3.7}$$

上式简记为

$$g(t) := x(t + \tau) \xleftrightarrow[\text{算法 3.1}]{\text{数值 CTFT}} G(\omega_k) = \exp(j\omega_k\tau) X(\omega_k), \tag{3.8}$$

其中 $\omega_k := \frac{k}{N}\omega_s$, $k \in \mathbb{Z}$.

式 (3.8) 与 CTFT 解析解的时移性质相同.

命题 3.3 算法 3.1 得到的数值 CTFT 能保持 CTFT 解析解的尺度变换和时移性质. ■

4 奇偶虚实性

考虑 $x(t)$ 的共轭信号 $g(t) := x^*(t)$. 用算法 3.1 得到的 $g(t)$ 的 CTFT 数值解

$$\begin{aligned}
 G(\omega)|_{\omega=\frac{k}{N}\omega_s} &= \exp\left(-j\frac{k}{N}\omega_s t_0\right) T_s \sum_{n=0}^{N-1} x^*(t_0 + nT_s) \exp\left(-jk\frac{2\pi}{N}n\right) \\
 &= \left[\exp\left(-j\frac{-k}{N}\omega_s t_0\right) T_s \sum_{n=0}^{N-1} x(t_0 + nT_s) \exp\left(-j(-k)\frac{2\pi}{N}n\right) \right]^* \\
 &= X^*(\omega)|_{\omega=\frac{-k}{N}\omega_s}, \quad k \in \mathbb{Z}.
 \end{aligned} \tag{3.9}$$



上式简记为

$$x^*(t) \xleftrightarrow[\text{算法 3.1}]{\text{数值 CTFT}} X^*(-\omega_k), \quad (3.9')$$

其中 $\omega_k := \frac{k}{N}\omega_s$, $k \in \mathbb{Z}$.

式 (3.9) 与 CTFT 解析解的相应性质相同.

特别地, 若 $x(t)$ 为实信号, 则 $X(\omega_k)$ 是共轭对称的: $X(\omega_k) = X^*(-\omega_k)$, 即

$$|X(\omega_k)| = |X(-\omega_k)|, \quad \arg(X(\omega_k)) = -\arg(X(-\omega_k)). \quad (3.10)$$

反之, 若 $X(\omega_k)$ 是共轭对称的, 则 $x(t)$ 是实的.

命题 3.4 对于实信号, 由算法 3.1 得到的数值 CTFT 的幅频特性是偶对称的, 相频特性是奇对称的. 反之, 对于共轭对称的频谱, 其由算法 3.2 得到的逆变换是实的. ■

3.3 结果和讨论

图 3.1 展示了原始音频信号, 及其时域尺度变换. 结果与预期相符.

图 3.2 示出了原始音频信号, 及其用算法 3.1 求出的数值 CTFT 的幅度谱和相位谱. 可见, 数值 CTFT 的尺度变换性质与解析 CTFT 的一致, 如同命题 3.3 指出的那样.

图 3.3 展示了对原始信号的数值 CTFT 用算法 3.2 求数值 ICTFT 的结果. 可见, 求出的 ICTFT 是实的, 与原始信号的差别基本上只是数值噪声, 这说明用算法 3.1 求数值 CTFT 与用算法 3.2 求数值 ICTFT 是互逆的.

图 3.4、图 3.5 分别给出了, 对原始信号用算法 3.1 求出的数值 CTFT 的幅度、相位谱, 再用算法 3.2 求数值 ICTFT 的结果, 可见它们都是实的, 如同命题 3.4 指出的那样. 它们与原始信号差别都非常大, 说明一个时域信号在频域上是由其幅度谱和相位谱共同决定的, 仅有幅度谱或仅有相位谱都不足以给出时域信号的完整信息.

图 3.6 给出了原始信号经过一个理想低通滤波器 (low-pass filter, LPF) 后的波形和与原始信号的差别. 理想 LPF 的截止频率选为 2.5 kHz, 这能保留原始信号的主要频率分量, 但会滤去部分高频分量. 从图 3.6 可看出, 滤波后的信号与原始信号的形状很相似, 但在许多点处实际上具有较大差异. 这说明, 虽然从原始信号的幅频特性上看, 这理想 LPF 将滤去的频率分量只占少数, 但已足以对波形产生一定的影响: 它使波形相对更平缓, 能保持原始信号的形状, 而对某些变化较剧烈的点起到一定的“削平”作用.



Original waveform and its time-domain transformation

Guorui Wei 516021910080

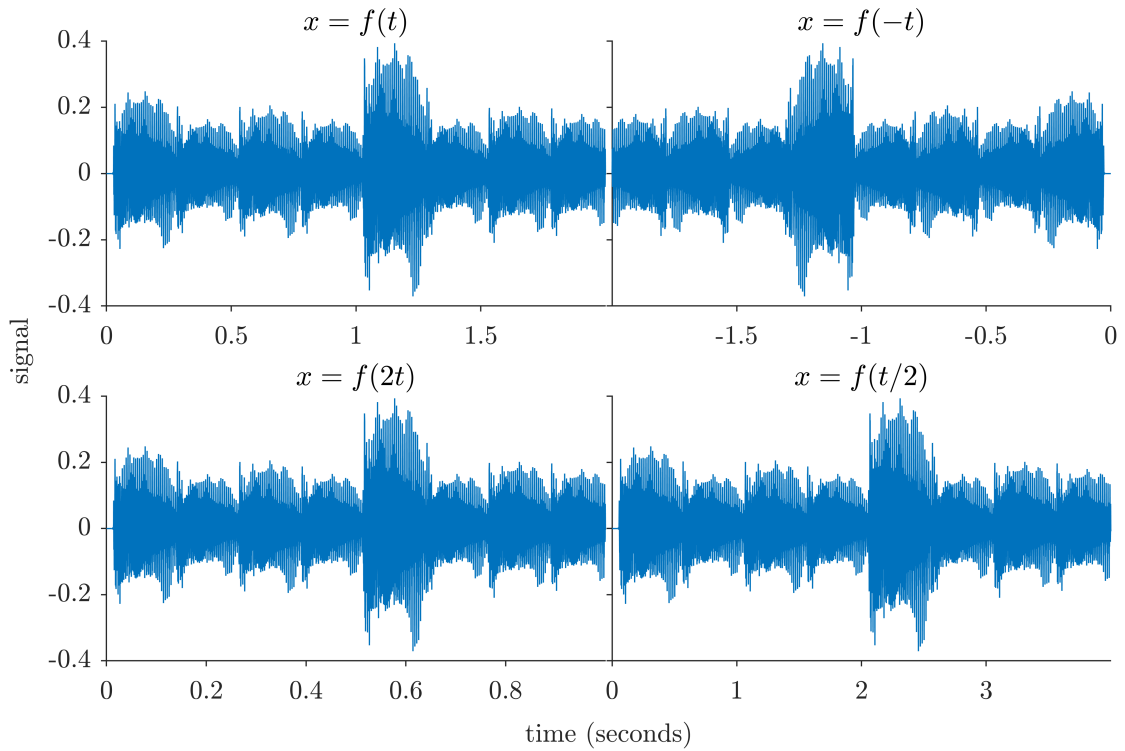
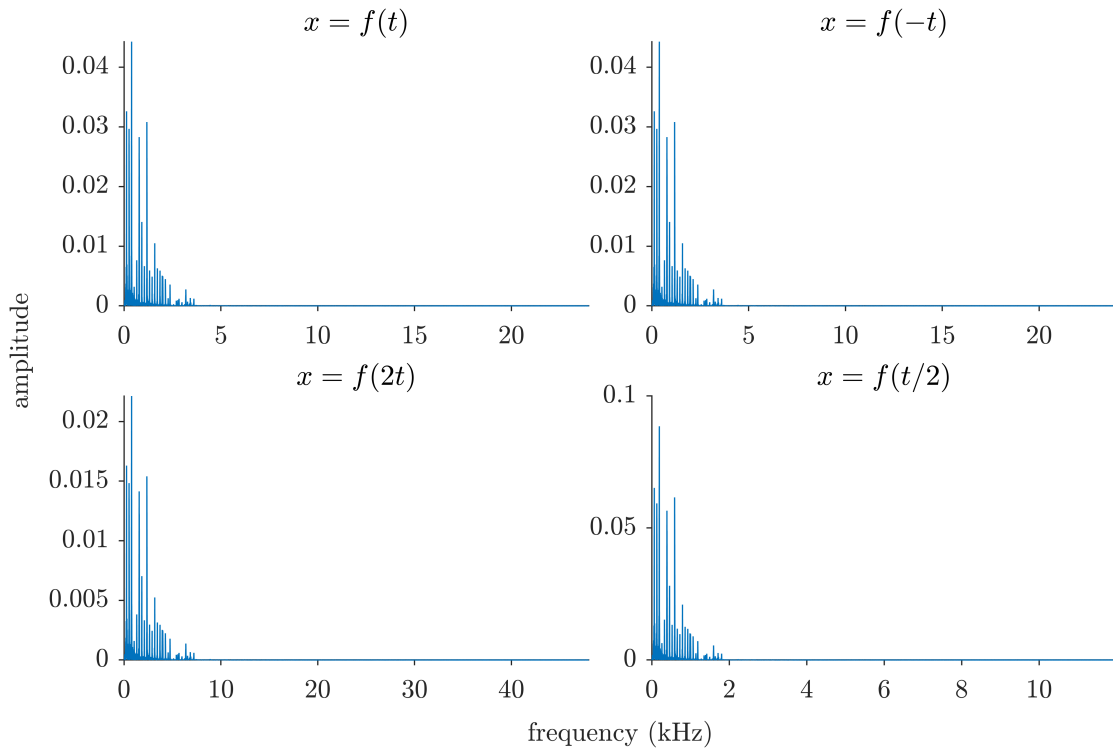


图 3.1 原始音频信号, 及其时域尺度变换. 注意横坐标轴刻度的差异.

CTFT (amplitude)

Guorui Wei 516021910080



(a)

**CTFT (phase)**

Guorui Wei 516021910080

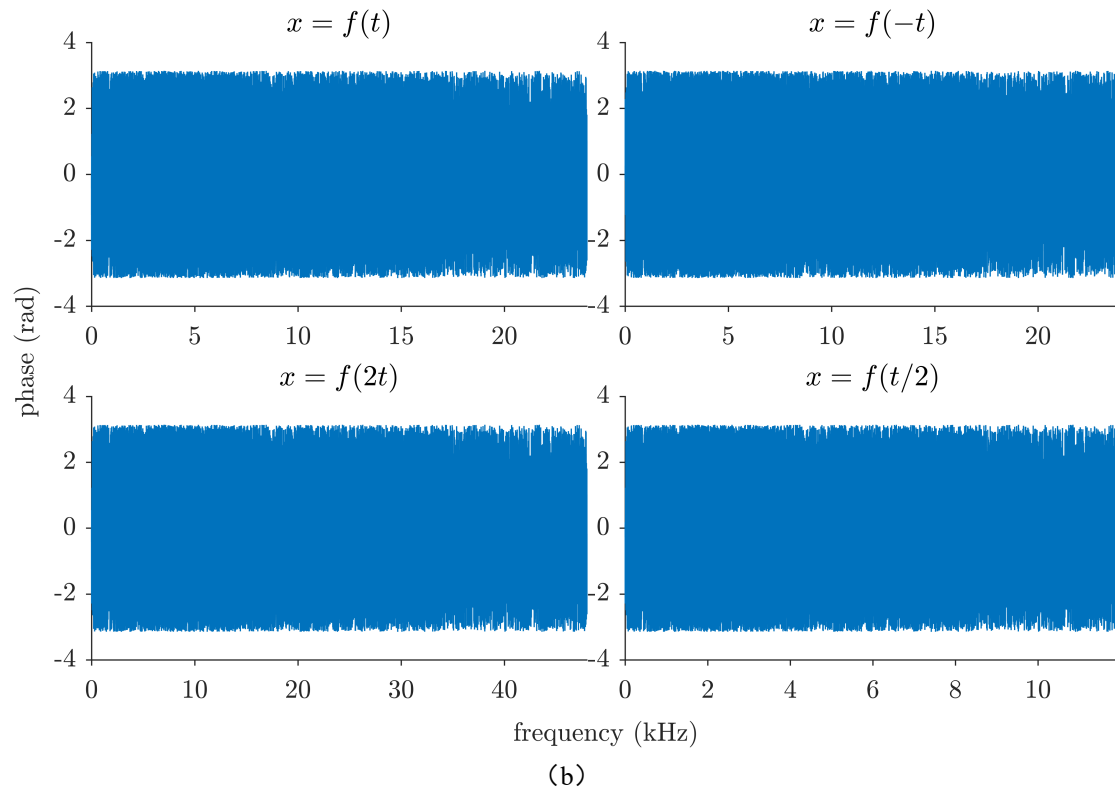


图 3.2 原始音频信号, 及其用算法 3.1 求出的数值连续时间 Fourier 变换 (continuous-time Fourier transform, CTFT) 的幅度谱 (a) 和相位谱 (b). 可见, 数值 CTFT 的尺度变换性质与解析 CTFT 的一致, 如同命题 3.3 指出的那样.



Inverse Fourier Transform

Guorui Wei 516021910080

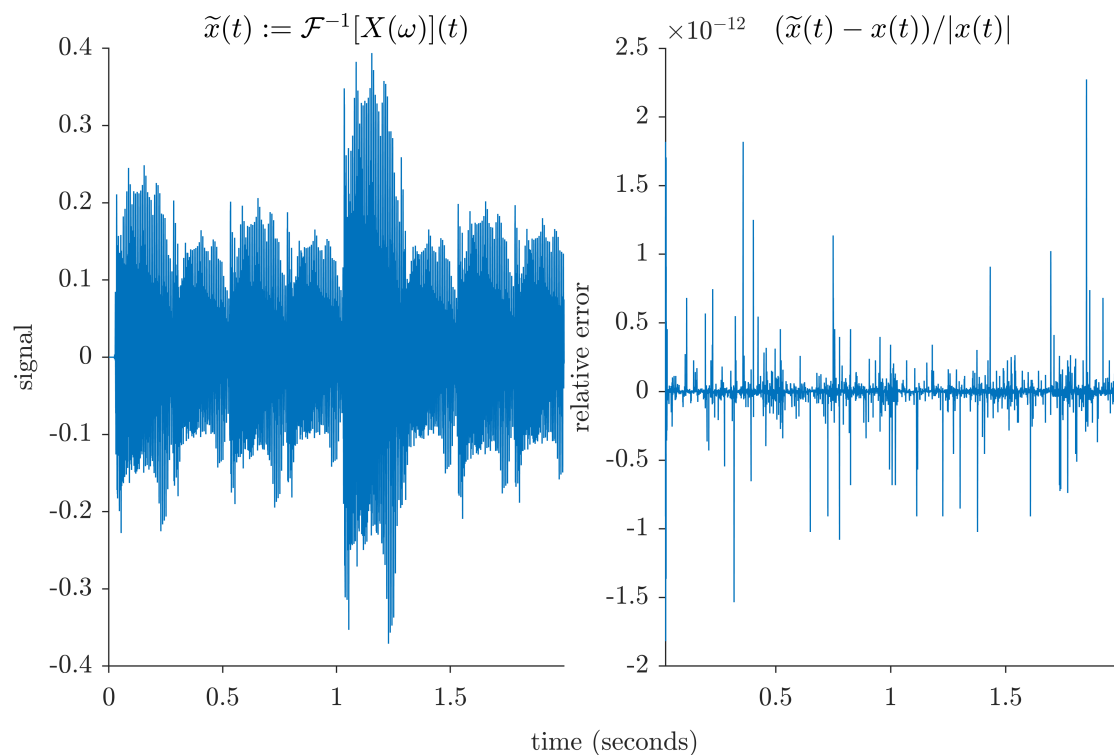


图 3.3 对原始信号的数值连续时间 Fourier 变换 (CTFT) 用算法 3.2 求数值连续时间 Fourier 逆变换 (inverse continuous-time Fourier transform, ICTFT) 的结果. 求出的 ICTFT 是实的 (与命题 3.4 一致), 波形示于左子图. 求出的数值 ICTFT 对原始信号的相对误差示于右子图, 可见这数值 ICTFT 与原始信号的差别基本上只是数值噪声, 说明用算法 3.1 求数值 CTFT 与用算法 3.2 求数值 ICTFT 是互逆的.



Inverse Fourier Transform of Magnitude Spectrum

Guorui Wei 516021910080

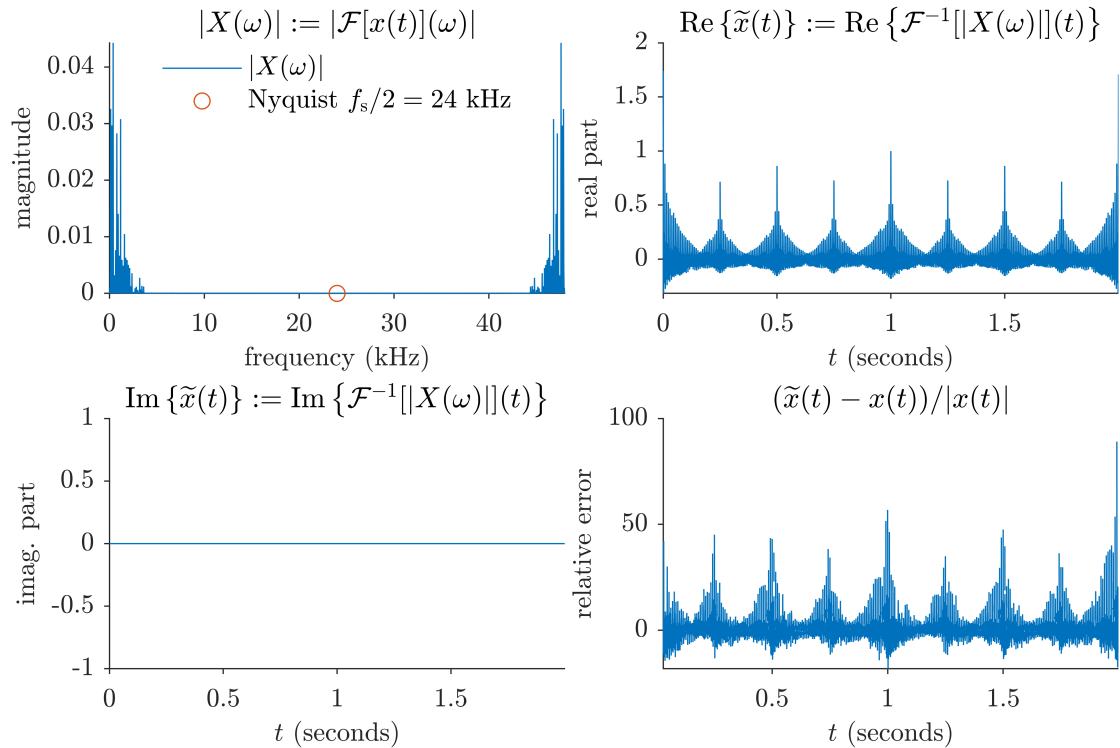


图 3.4 对原始信号用算法 3.1 求出的数值 CTFT 的幅度谱, 再用算法 3.2 求数值 ICTFT 的结果. 左上图是对原始信号用算法 3.1 求出的数值 CTFT 的幅度谱. 对这 CTFT 幅度谱用算法 3.2 求出的数值 ICTFT 的实部和虚部分别示于右上图和左下图, 可见这数值 ICTFT 是实的 (与命题 3.4 一致). 这数值 ICTFT 对原始信号的相对误差示于右下图, 在实际计算中跳过了那些使得原始信号的绝对值很小的时刻; 可见与原始信号差别非常大.



Inverse Fourier Transform of Phase Spectrum

Guorui Wei 516021910080

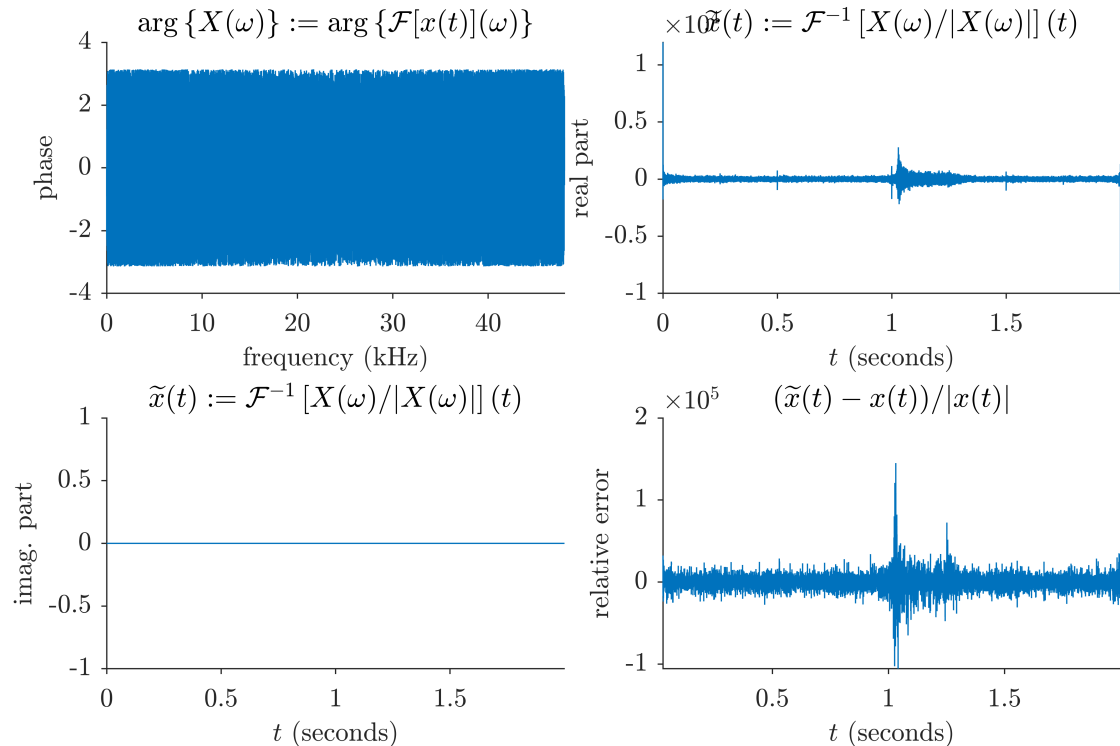


图 3.5 对原始信号用算法 3.1 求出的数值 CTFT 的相位谱, 再用算法 3.2 求数值 ICTFT 的结果. 左上图是对原始信号用算法 3.1 求出的数值 CTFT 的相位谱. 对这 CTFT 相位谱用算法 3.2 求出的数值 ICTFT 的实部和虚部分别示于右上图和左下图, 可见这数值 ICTFT 是实的 (与命题 3.4 一致). 这数值 ICTFT 对原始信号的相对误差示于右下图, 在实际计算中跳过了那些使得原始信号的绝对值很小的时刻; 可见差别非常大.

ideal-LPF-filtered waveform

Guorui Wei 516021910080

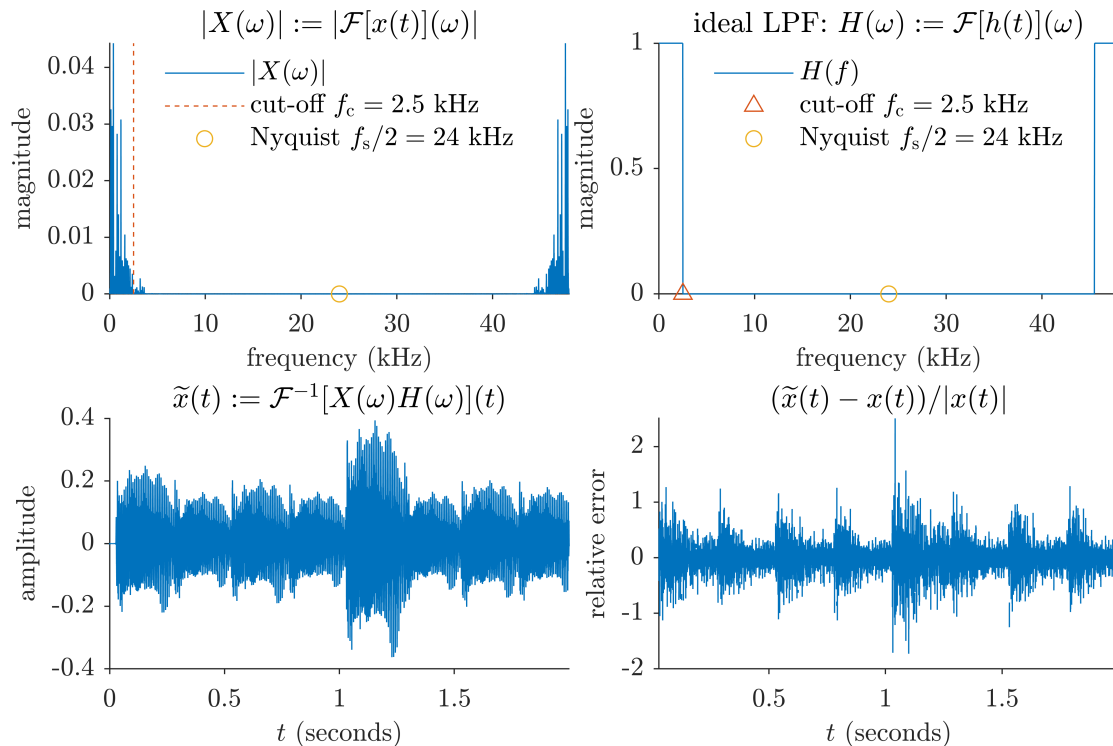


图 3.6 原始信号经过一个理想低通滤波器 (low-pass filter, LPF) 后的波形和与原始信号的差别. 左上图展示了对原始信号用算法 3.1 求出的数值 CTFT 的幅度谱, 理想 LPF 的截止频率 (2.5 kHz), 以及原始音频信号的采样率对应的 Nyquist 频率 ($48/2 = 24$ kHz). 右上图展示了理想 LPF 的频率响应, 它是实的. 左下图展示了原始信号经过一个理想低通滤波器 (low-pass filter, LPF) 后的波形, 可见与原始信号形状很相似. 右下图展示了滤波后的信号对原始信号的相对误差, 在实际计算中跳过了那些使得原始信号的绝对值很小的时刻; 可见, 虽然滤波后的信号的形状与原始信号很相似, 但在许多点处差异较大.

3.4 项目小结

本项目利用 N 点 DFT 与 CTFT 的关系 (2.24), 系统地建立了用 DFT 数值计算 CTFT 的算法 3.1. 推导了数值 CTFT 的数值反变换 (3.2), 从而建立了用 IDFT 数值计算 ICTFT 的算法 3.2. 推导了数值 CTFT 的尺度变换性质 (3.6), 时移性质 (3.8) 和奇偶虚实性 (3.9), 发现这些性质与解析 CTFT 的一致.

本项目制作了一段简单的音乐, 通过对这音频信号施加时域尺度变换、用算法 3.1 计算原始信号及其时域尺度变换的数值 CTFT、用算法 3.2 计算原始音频信号的数值 CTFT 的数值 ICTFT, 验证了前述算法和性质. 通过分别对原始音频信号的数值 CTFT 的幅度谱、相位谱单独做数值 ICTFT, 将结果与原始信号作比较, 说明一个时域信号在频域上是由其幅度谱和相位谱共同决定的, 仅有幅度谱或仅有相位谱都不足以给出时域信号的完整信息. 设计了一个理想低通 LPF, 将原始音频信号用这 LPF 滤波, 发现这 LPF 可保持原始信号的形状, 且起到一定的“平滑”作用.



4 项目 2：语音采样器 (Speech Sampler)

4.1 项目简介

采集某个人的声音作为一段连续时间信号，使用不同的采样频率对其进行若干次采样，得到若干份离散时间信号。(推荐采样频率：44 kHz, 22 kHz, 11 kHz, 5.5 kHz, 2.75 kHz)

对上述离散时间信号分别进行重构，得到相应的连续时间信号，分析不同采样频率对重构质量的影响，并计算重构误差。

分析“Speech Sampler”的基本设计思路和原理，并用编程语言实现之。

4.2 数据和方法

4.2.1 连续时间信号

为了突出研究主题——采样与重建，本文使用仅含 3 个单音频率分量的连续时间信号 (图 4.1)

$$x(t) := \sum_{i=1}^3 \sin(2\pi f_i t + \varphi_i), \quad (4.1)$$

其中 $f_1 = 10$ kHz, $f_2 = 5$ kHz, $f_3 = 2.5$ kHz, $\varphi_1 = \pi/6$, $\varphi_2 = \pi/2$, $\varphi_3 = 5\pi/6$.

这种信号可通过计算机软件 (例如 Matlab) 方便地生成。

original signal (periodic)

Guorui Wei 516021910080

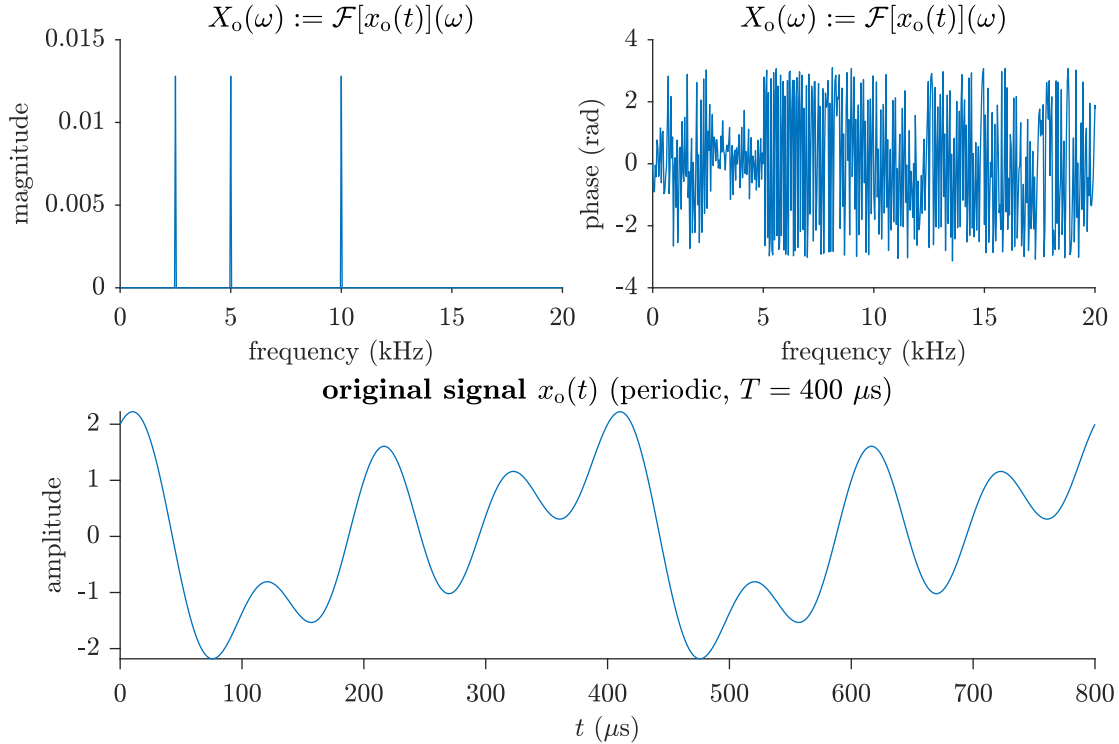


图 4.1 本项目使用的连续时间信号 $x(t)$. 该信号仅含 3 个等幅且初相不同的单音频率分量, 即 $x(t) := \sum_{i=1}^3 \sin(2\pi f_i t + \varphi_i)$, $f_1 = 10$ kHz, $f_2 = 5$ kHz, $f_3 = 2.5$ kHz, $\varphi_1 = \pi/6$, $\varphi_2 = \pi/2$, $\varphi_3 = 5\pi/6$. 因此 $x(t)$ 具有 $400 \mu\text{s}$ 周期. 本图展示了 $x(t)$ 在两个周期上的图像.

4.2.2 冲激串采样与理想带限内插

对连续时间信号 $x(t)$ 执行如图 4.2 所示的冲激串采样和重建流程. 原始信号 $x(t)$ 与周期采样信号

$$p(t) := \sum_n \delta(t - nT_s) = \frac{1}{T_s} \sum_k e^{jk\omega_s t} \xleftrightarrow{\text{CTFT}} P(\omega) = \omega_s \sum_k \delta(\omega - k\omega_s), \quad (4.2)$$

相乘, 得采样后信号 $x_p(t)$, 其频谱

$$X_p(\omega) := \mathcal{F}[x_p(t)] = \mathcal{F}[x(t)p(t)] = \frac{1}{2\pi} X(\omega) * P(\omega) = \frac{1}{T_s} \sum_k X(\omega - k\omega_s). \quad (4.3)$$

可见, $X_p(\omega)$ 就是 $X(\omega)$ 的等幅周期拓展.

若 $x(t)$ 是带限(band-limited)的, 即 $x(t)$ 的频谱满足

$$X(\omega) := \mathcal{F}[x(t)](\omega) = 0, \quad \forall \omega: |\omega| > \omega_m, \quad (4.4)$$

则只需将取样后的信号 $x_p(t)$ 通过一个截止频率为 ω_c , $\omega_m < \omega_c < \omega_s - \omega_m$, 幅度为 T_s 的理想低通滤波器 (频率响应记为 $H(\omega)$), 就可恢复出原始信号 $x(t)$. 理想低通滤波器的冲激响应

$$h(t) = \mathcal{F}^{-1}[H(\omega)] = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} T_s e^{j\omega t} d\omega = \frac{2\omega_c}{\omega_s} \text{sinc} \frac{\omega_c t}{\pi}, \quad (4.5)$$

上式中的 sinc 函数定义为

$$\text{sinc } t := \begin{cases} 1, & t = 0, \\ \frac{\sin(\pi t)}{\pi t}, & t \neq 0. \end{cases} \quad (4.6)$$

得到的重建信号

$$x_r(t) = x_p(t) * h(t) = \sum_n x(nT_s)\delta(t - nT_s) * h(t) = \sum_n x(nT_s)h(t - nT_s). \quad (4.7)$$

若取 $\omega_c = \omega_s/2 \in (\omega_m, \omega_s - \omega_m)$, $\omega_s > 2\omega_m$, 则有

$$x_r(t) = \sum_n x(nT_s) \operatorname{sinc}\left(\frac{t}{T_s} - n\right). \quad (4.8)$$

特别地,

$$x_r(nT_s) = \sum_m x(mT_s) \operatorname{sinc}(n - m) = x(nT_s), \quad \forall n \in \mathbb{Z}. \quad (4.9)$$

可见, 重建信号 $x_r(t)$ 经过所有采样点 $(nT_s, x(nT_s))$, 可将 $x_r(t)$ 视作在插值节点 $(nT_s, x(nT_s))$, $n \in \mathbb{Z}$ 下, 以 sinc 函数作为插值基函数的一个插值函数, 称为理想带限内插 (ideal band-limited interpolation).

定理 4.1 (ideal band-limited interpolation) 设带限信号 $x(t)$ 满足 (4.4), 其等间隔采样值 $x(nT_s)$. 若采样频率 $\omega_s := 2\pi/T_s$ 满足 $\omega_s > 2\omega_m$, 则以这些采样时刻作插值节点, 以 sinc 函数作为插值基函数作理想带限内插 (4.8), 可精确重建 $x(t)$. ■

注 4.2 (时域有限信号的重建) 重建公式 (4.8) 是无穷级数, 需要无限个采样值, 一般难以实现. 对于时域有限信号, 因其除一个有限时间区间外恒为零, 故只需在其非恒零区间上等时间间隔采样, 这时 (4.8) 成为有限项求和. 但时域有限信号通常是频域无限的, 通常不满足定理 4.1 的条件, 这时通过 (4.8) 得到的重建信号只是原始信号的近似. ■

对于时域无限的原始信号, 通常只能获得该信号在一个有限时间区间上的等间隔采样值. 为了使用重建公式 (4.8), 可以规定在实际取样区间外的“虚拟”样值都为零. 这时, 重建公式 (4.8) 是对这样一个信号的近似: 它在采样时间区间内同原始信号相同, 在采样时间区间外为零 (或至少在“虚拟”采样时刻为零); 而它通常不是频域有限的, 即使原始信号是频域有限的. 这样, 重建公式 (4.8) 可认为是在采样时间区间内对原始信号的一个近似, 而在采样时间区间外不具有重建意义. 从插值的角度看, 重建公式 (4.8) 可用于内插 (interpolation), 而不用外推 (extrapolation), 故称为理想带限内插.

注 4.3 (时域无限信号在有限区间上的重建) 对于时域无限信号, 可以利用其在有限时间区间上的等时间间隔采样值和重建公式 (4.8), 在该时间区间上作内插 (不宜用于外推). 这种“截断”方式通常会在原始信号的基础上引入高频分量, 导致定理 4.1 的条件不再被满足. 因此, 即使原始信号是频域有限的, 通常也不能通过 (4.8) 在采样时间区间上精确重建原始信号. ■

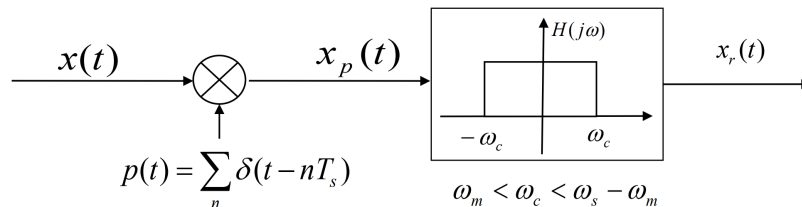


图 4.2 理想采样与重建原理图. $x(t)$ 是带限信号 (band-limited signal), 当 $|\omega| > \omega_m$ 时, $x(t)$ 的频谱 $X(\omega) := \mathcal{F}[x(t)](\omega) = 0$. $p(t)$ 是采样函数 (sampling function), 在冲激串采样中取为周期冲激串, 取样周期为 T_s , 即取样频率 $\omega_s := 2\pi/T_s$. 取样后的信号 $x_p(t)$ 通过一个截止频率为 ω_c , 幅度为 T_s 的理想低通滤波器, 得到重建信号 $x_r(t)$. 若 $\omega_m < \omega_c < \omega_s - \omega_m$, 则有 $x_r(t) = x(t)$, $\forall t$. 图源: 上海交通大学 2022 年春季学期 ICE2301 课程教学用课件“signal&system(v3.1)-ch4-FT-频域分析.pdf”第 66 页 (刘伟, 2022).



4.2.3 基于理想带限内插的信号重建

设有时域有限的连续时间信号 $x(t) \in C_c[t_0, t_0 + \tau)$, 其等时间间隔样值

$$x(t_0 + nT_s), \quad n = 0, \dots, N-1,$$

其中 $NT_s = \tau$. 下面建立其理想带限内插.

记 $y(t) := x(t + t_0)$, 则 $y(t) \in C_c[0, \tau)$, 其等时间间隔样值

$$y(nT_s) := x(t_0 + nT_s), \quad n = 0, \dots, N-1.$$

于是 $y(t)$ 的理想带限内插 (4.8) 成为

$$y_r(t) = \sum_{n=-\infty}^{+\infty} y(nT_s) \operatorname{sinc}\left(\frac{t}{T_s} - n\right) = \sum_{n=0}^{N-1} x(t_0 + nT_s) \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right). \quad (4.10)$$

若 $y(t)$ 满足定理 4.1 的条件 (实际中很难满足, 因为时域有限信号通常是频域无限的), 则 $x(t)$ 也满足定理 4.1 的条件 (因为 $x(t), y(t)$ 具有相同的幅频特性), 从而

$$x_r(t) = x(t) = y(t - t_0) = y_r(t - t_0) = \sum_{n=0}^{N-1} x(t_0 + nT_s) \operatorname{sinc}\left(\frac{t - (t_0 + nT_s)}{T_s}\right). \quad (4.11)$$

为此, 规定

$$x_r(t) := \sum_{n=0}^{N-1} x(t_0 + nT_s) \operatorname{sinc}\left(\frac{t - (t_0 + nT_s)}{T_s}\right) \quad (4.12)$$

为时域有限的连续时间信号 $x(t) \in C_c[t_0, t_0 + \tau)$ 的**理想带限内插**.

定义连续时间信号 $x(t)$ 的样值序列 \mathbf{x} 和取样函数序列 $\mathbf{s}(t)$:

$$\begin{aligned} \mathbf{x} &:= (x_i)_{N \times 1} := (x(t_0 + iT_s))_{N \times 1}, \quad i = 0, \dots, N-1, \\ \mathbf{s}(t) &:= (s_i(t))_{N \times 1} := \left(\operatorname{sinc}\left(\frac{t - (t_0 + iT_s)}{T_s}\right) \right)_{N \times 1}, \quad i = 0, \dots, N-1, \end{aligned} \quad (4.13)$$

即

$$\begin{aligned} \mathbf{x} &:= [x(t_0 + 0T_s), x(t_0 + 1T_s), \dots, x(t_0 + (N-1)T_s)]^T, \\ \mathbf{s}(t) &:= \left[\operatorname{sinc}\left(\frac{t - (t_0 + 0T_s)}{T_s}\right), \dots, \operatorname{sinc}\left(\frac{t - (t_0 + (N-1)T_s)}{T_s}\right) \right]^T. \end{aligned} \quad (4.13')$$

则 (4.12) 可写成

$$x_r(t) := \mathbf{s}^T(t) \mathbf{x}. \quad (4.12')$$

现在用等时间间隔样值 \mathbf{x} 理想带限内插到 $t = t_k$, $k = 0, \dots, M-1$ 上, 试图重建原始信号 $x(t)$. 定义重建信号序列

$$\mathbf{x}_r := (x_r(t_k))_{M \times 1}, \quad k = 0, \dots, M-1,$$

则由 (4.12') 得

$$\mathbf{x}_r = \begin{bmatrix} \mathbf{s}^T(t_0) \\ \vdots \\ \mathbf{s}^T(t_{M-1}) \end{bmatrix} \mathbf{x}. \quad (4.14)$$

上式这种矩阵形式适合计算机编程实现. 若定理 4.1 的条件被满足, 则 \mathbf{x}_r 和原始信号在相应时刻的值相等.

4.2.4 信号重建质量的评价

由 (4.9) 知道, 理想带限内插总是能够精确恢复原始信号在取样时刻的值. 因此, 为了评价信号重建的质量, 应考察重建公式 (4.8) 对非取样时刻原始信号的恢复情况. 注 4.3 指出, 重建公式 (4.8) 不宜用于外推, 故不必考察 (4.8) 在取样时间区间外的重建性能.



基于上述考虑,本项目按下述方式评价信号重建质量.在采样时间区间内,等间隔地选定一组重建时刻,重建时间间隔远小于且整除采样时间间隔,以使重建时刻包含采样时刻.计算**重建相对误差**

$$e_r(t_r) := \frac{x_r(t_r) - x(t_r)}{|x(t_r)|}, \quad (4.15)$$

其中 t_r 是重建时刻, $x_r(t)$ 是利用理想带限内插重建公式 (4.8) 对原始信号 $x(t)$ 在采样时间区间上的重建信号. 定义 (4.15) 在分母为零或接近零时不宜使用, 故在实际计算中跳过了那些使得 (4.15) 的分母很小的点.

4.3 结果和讨论

4.3.1 重建的信号

对仅含 10, 5, 2.75 kHz 三个频率分量的原始信号 (4.1), 从初始时刻 0 秒 (含) 起, 分别以采样频率 44、22、11、5.5、2.75 kHz 等间隔取样, 获得 2253、1127、564、282、141 个采样值. 在采样时间区间内, 以重建频率 320 kHz, 等时间间隔地选取 16385 个重建时刻, 使用理想带限内插公式 (4.8) 获得重建信号在重建时刻上的值. 用 (4.15) 计算重建相对误差. 重建信号及其频谱和重建误差示于图 4.3.

从图 4.3(a)可见, 当采用 44 kHz 的采样频率时, 在取样时间区间内, 理想带限内插公式 (4.8) 可很好地重建原始信号. 重建信号的频谱显示了与原始信号的相同的三个单音分量. 重建相对误差的绝对值在取样时间区间中心附近的一个半径较大的邻域内维持低值 (小于 0.01%), 而在区间边缘相对增大, 最大超过 48%.

从图 4.3(b)可见, 当采用 22 kHz 的采样频率时, 在取样时间区间内, 理想带限内插公式 (4.8) 可较好地重建原始信号, 但重建误差比 44 kHz 采样频率的大. 重建信号的频谱显示了与原始信号的相同的三个单音分量, 且在 Nyquist 频率 11 kHz 附近显示了一个略微突出但明显可见的频率分量. 重建相对误差的绝对值在取样时间区间中心附近的一个半径较大的邻域内维持较低值 (小于 1%), 而在区间边缘增大, 最大超过 94%.

从图 4.3(c)可见, 当采用 11 kHz 的采样频率时, 在取样时间区间内, 理想带限内插公式 (4.8) 不能重建原始信号. 从重建信号的频谱可见, 原始信号中 5 kHz 和 2.5 kHz 两个单音分量得到保持, 但 10 kHz 的单音分量被混叠成 0.996 kHz (理论值 $11 - 10 = 1$ kHz) 的单音分量; 且在 Nyquist 频率 5.5 kHz 附近显示了一个略微突出但明显可见的频率分量.

从图 4.3(d)可见, 当采用 5.5 kHz 的采样频率时, 在取样时间区间内, 理想带限内插公式 (4.8) 不能重建原始信号. 从重建信号的频谱可见, 原始信号中 2.5 kHz 的单音分量得到保持, 但 10 kHz 的单音分量被混叠成 0.98 kHz (理论值 $5.5 \times 2 - 10 = 1$ kHz) 的单音分量, 5 kHz 的单音分量被混叠成 0.49 kHz (理论值 $5.5 - 5 = 0.5$ kHz) 的单音分量; 且在 Nyquist 频率 2.75 kHz 附近显示了一个略微突出但明显可见的频率分量.

从图 4.3(e)可见, 当采用 2.75 kHz 的采样频率时, 在取样时间区间内, 理想带限内插公式 (4.8) 不能重建原始信号. 从重建信号的频谱可见, 原始信号中 10 kHz 的单音分量被混叠成 0.94 kHz (理论值 $2.75 \times 4 - 10 = 1$ kHz) 的单音分量, 5 kHz 的单音分量被混叠成 0.47 kHz (理论值 $2.75 \times 2 - 5 = 0.5$ kHz) 的单音分量, 2.5 kHz 的单音分量被混叠成 0.23 kHz (理论值 $2.75 - 2.5 = 0.25$ kHz) 的单音分量; 且在 Nyquist 频率 1.375 kHz 附近显示了一个略微突出但明显可见的频率分量.

4.3.2 讨论

对图 4.3 所示的实验结果, 可得出以下观察.



1. 即使采样频率高于原始信号对应的 Nyquist 频率,也不能通过理想带限内插公式(4.8)在采样时间区间内精确重建原始信号. 这是因为,使用有限的采样时间区间,意味着重建的实际上是作为原始信号一部分的一个时域有限信号,而时域有限信号通常是频域无限的,不满足定理 4.1 的条件. 这与注 4.3 的描述一致.

2. 当采样频率高于原始信号对应的 Nyquist 频率时,在采样时间充分长(至少覆盖原始信号的几个周期)的情况下,可在采样时间区间内较好地重建原始信号,且采样频率越高、采样时间越长,重建质量越好.

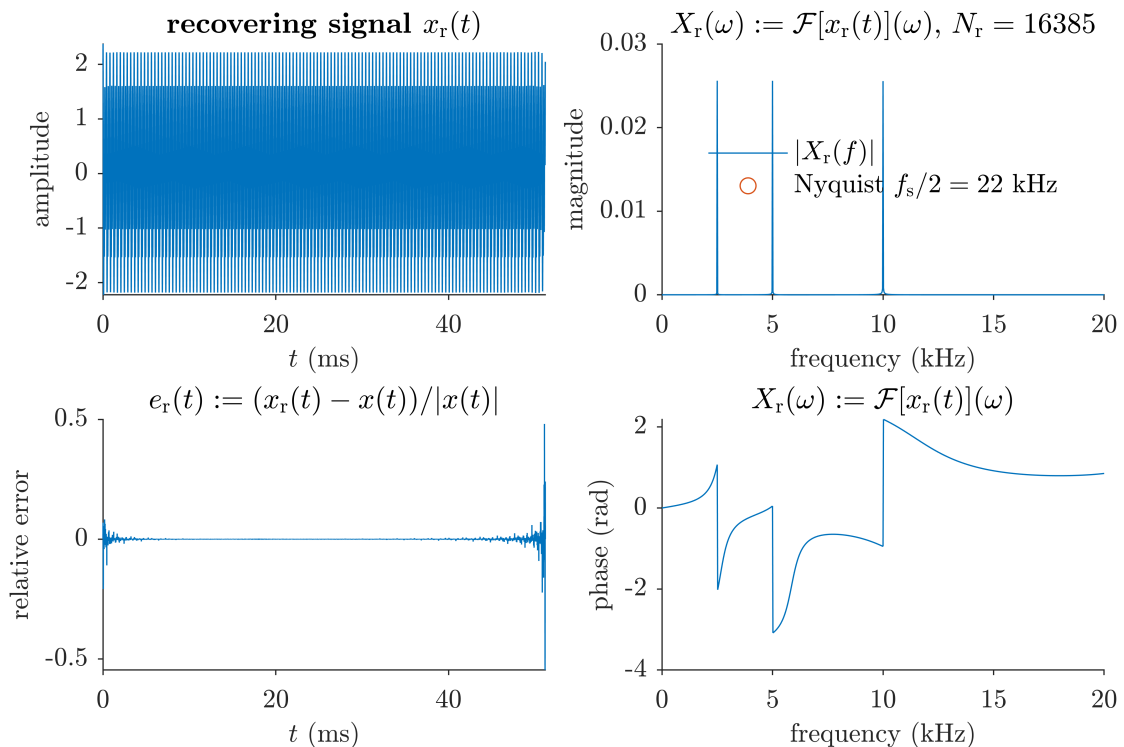
3. 当采样频率高于原始信号对应的 Nyquist 频率时,重建误差在采样时间区间的中心附近较小,而在接近区间边缘处升高,类似 Gibbs 现象.

4. 当采样频率低于原始信号对应的 Nyquist 频率时,不能重建原始信号. 这时,原始信号在采样-重建过程中会发生**频谱混叠**(aliasing),较高频率的分量会“混叠”成较低频率的分量,导致不能从采样值恢复原始信号.

5. 利用时域无限信号在有限采样时间区间上的等间隔样值,通过理想带限内插公式(4.8)获得的重建信号,与原始信号相比,其频率在采样频率的二分之一附近的频率分量可能会略有增强. 原因可能是,通过有限采样时间区间上的样值按(4.8)恢复原始信号,隐含采样时间区间外的样值全为零,相当于在采样时间区间边缘引入了一个从非零到零的跳变,这对应一个频率约等于采样频率的二分之一的分量.

ideal band-limited interpolation ($f_s = 44 \text{ kHz}$, $N_s = 2253$)

Guorui Wei 516021910080

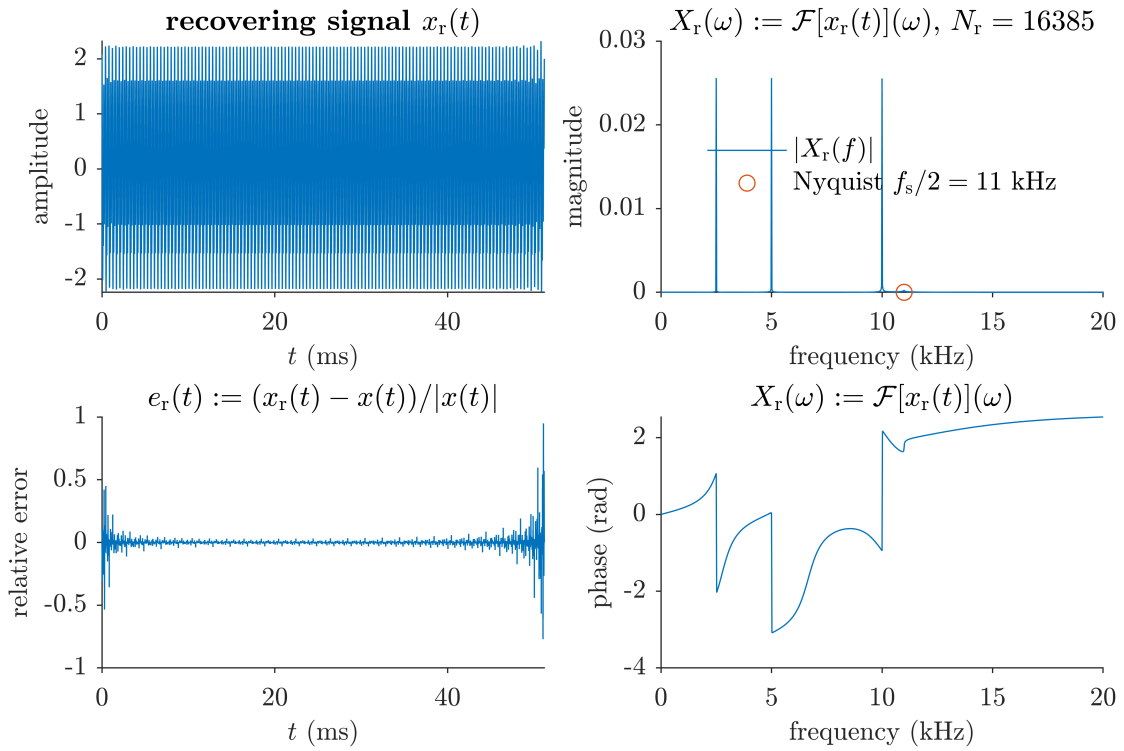


(a)



ideal band-limited interpolation ($f_s = 22$ kHz, $N_s = 1127$)

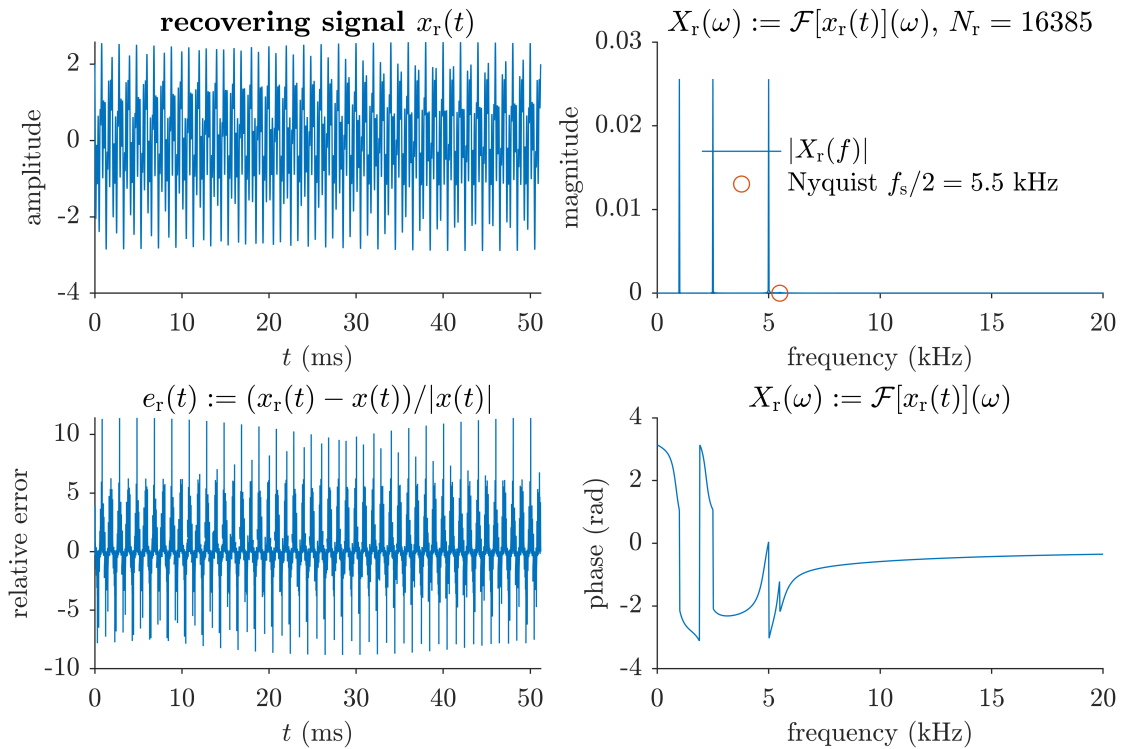
Guorui Wei 516021910080



(b)

ideal band-limited interpolation ($f_s = 11$ kHz, $N_s = 564$)

Guorui Wei 516021910080

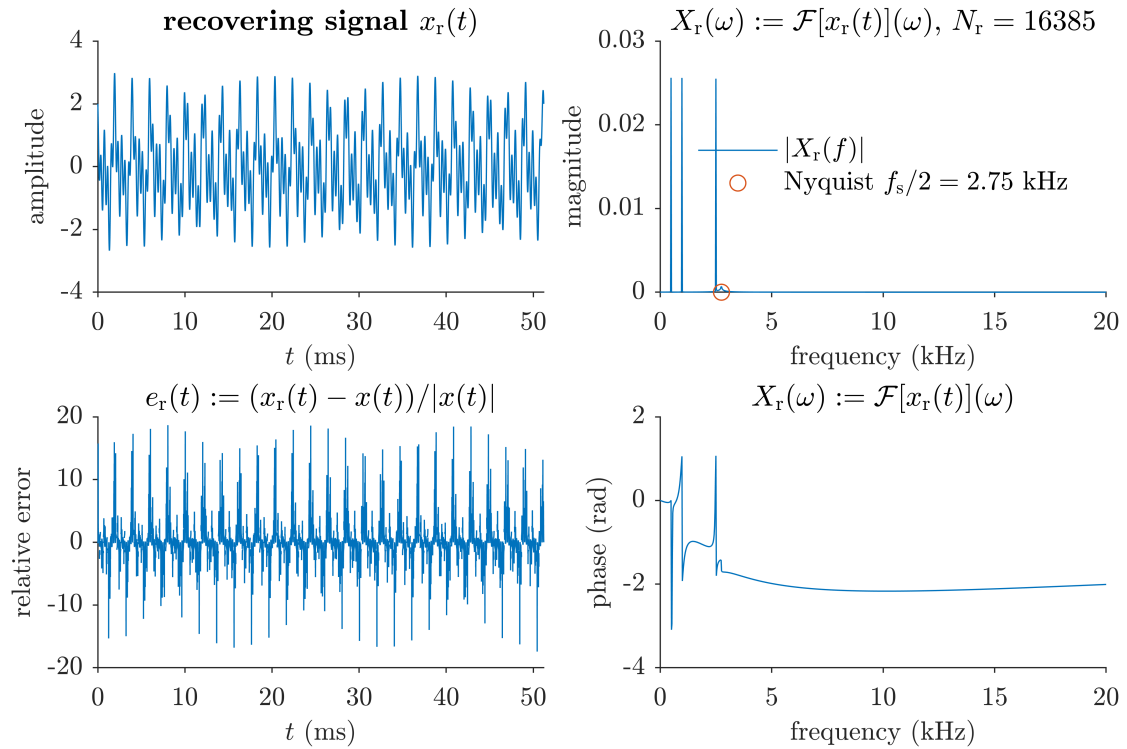


(c)



ideal band-limited interpolation ($f_s = 5.5$ kHz, $N_s = 282$)

Guorui Wei 516021910080



(d)

ideal band-limited interpolation ($f_s = 2.75$ kHz, $N_s = 141$)

Guorui Wei 516021910080

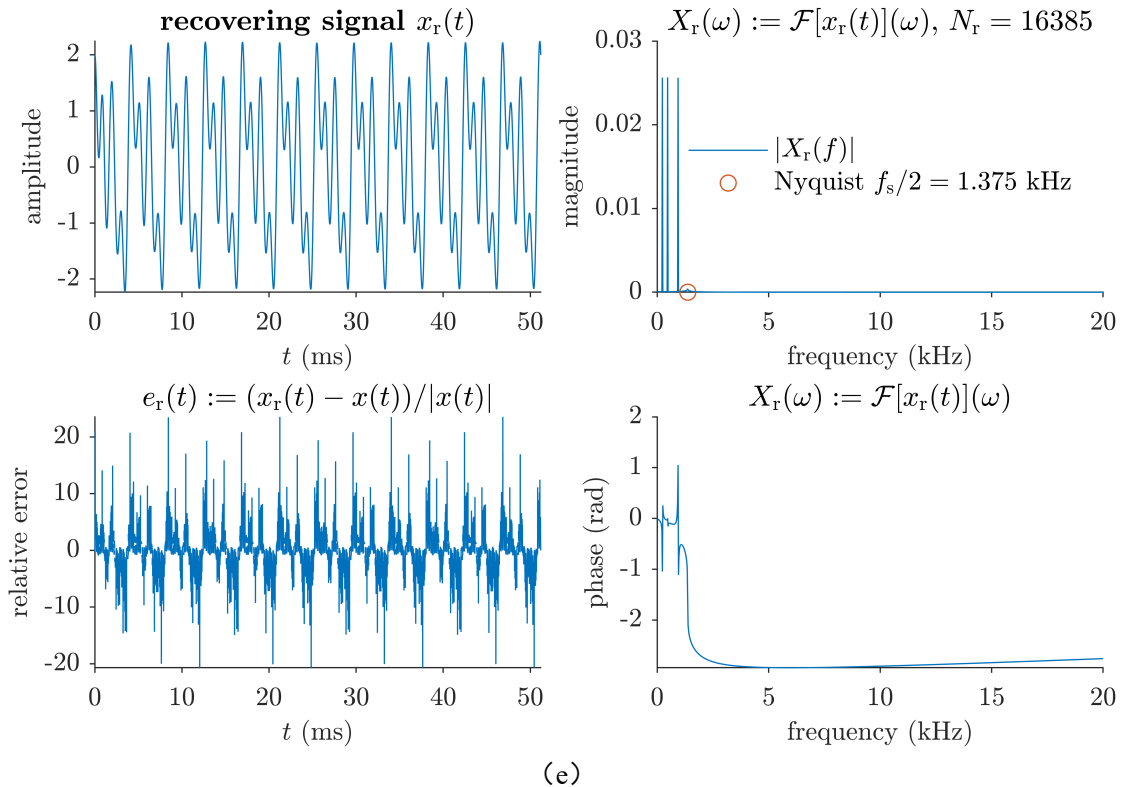


图 4.3 重建的信号. 对原始信号 (4.1) 以采样频率 (a) 44 kHz, (b) 22 kHz, (c) 11 kHz, (d) 5.5 kHz, (e) 2.75 kHz 等间隔取样, 获得 (a) 2253, (b) 1127, (c) 564, (d) 282, (e) 141 个采样值. 在采样时间区间内, 以重建频率 320 kHz, 等时间间隔地选取 16385 个重建时刻, 使用理想带限内插公式 (4.8) 计算重建信号在重建时刻上的值. 用 (4.15) 计算重建相对误差.

4.4 项目小结

本项目通过对理想冲激串采样-重建流程的理论分析, 建立了基于理想带限内插的信号重建公式 (4.8). 利用该公式, 在不同的采样频率下, 对仅含三个单音频率分量的简单连续时间周期信号 (4.1) 进行采样和重建, 得出了以下结论:

(1) 即使采样频率高于原始信号对应的 Nyquist 频率, 也不能通过理想带限内插公式 (4.8) 在采样时间区间内精确重建原始信号.

(2) 当采样频率高于原始信号对应的 Nyquist 频率时, 在采样时间充分长的情况下, 可在采样时间区间内较好地重建原始信号, 且采样频率越高、采样时间越长, 重建质量越好. 重建误差在采样时间区间的中心附近较小, 而在接近区间边缘处升高.

(3) 当采样频率低于原始信号对应的 Nyquist 频率时, 不能重建原始信号. 这时, 原始信号在采样-重建过程中会发生**频谱混叠** (aliasing), 较高频率的分量会“混叠”成较低频率的分量, 导致不能从采样值恢复原始信号.

(4) 利用时域无限信号在有限采样时间区间上的等间隔样值, 通过理想带限内插公式 (4.8) 获得的重建信号, 与原始信号相比, 其频率在采样频率的二分之一附近的频率分量可能会略有增强, 这是因为在有限采样区间上基于公式 (4.8) 的采样-重建过程隐含着信号在采样区间边缘存在一个从非零到零的突变; 这是重建误差的来源之一.



References



Appendix A MATLAB 源代码

本文使用的 Matlab 源代码和文档发布于 https://grwei.github.io/SJTU_2021-2022-2_ICE2301/.

A.1 proj1.m

```
1 %% proj1.m
2 % Description: MATLAB code for course project (ICE2301, 2022 Spring)
3 % Author: Guorui Wei (危国锐) (313017602@qq.com; weiguorui@sjtu.edu.cn)
4 % Student ID: 516021910080
5 % Created: 2022-05-20
6 % Last modified: 2022-05-27
7
8 %% Initialize project
9
10 clc; clear; close all
11 init_env();
12
13 %% Load data.
14
15 audio_path = '../data/original_simple2.wav';
16 info = audioinfo(audio_path);
17 [x,F_s] = audioread(audio_path); % F_s: sample rate (Hz)
18 % sound(x,F_s)
19 x_ori = x(:,1); % only one single channel will be analysed
20 t_ori = 0:1/F_s:info.Duration;
21 t_ori = t_ori(1:end-1).';
22 t_1 = -t_ori;
23 t_2 = t_ori/2;
24 t_3 = t_ori*2;
25
26 %% Continuous-time Fourier transform, approximated by DFT.
27
28 %% Create Fig.2
29
30 t_fig_ampl = figure('Name',"Fig.2a CTFT (amplitude) of
    original_wave_and_its_time-domain_transformation");
31 t_fig pha = figure('Name',"Fig.2b CTFT (phase) of orginal_wave_and_its_time-
    domain_transformation");
32 t_TCL_ampl =
    tiledlayout(t_fig_ampl,2,2,"TileSpacing","tight","Padding","tight");
```



```
33 t_TCL_pha =
    tiledlayout(t_fig_pha,2,2,"TileSpacing","tight","Padding","tight");
34 %
35 [omega_ori,CTFT_ori] =
    CTFT_fig(t_ori,x_ori,nexttile(t_TCL_ampl,1),nexttile(t_TCL_pha,1),"$x =
    f(t)$");
36 CTFT_fig(t_1,x_ori,nexttile(t_TCL_ampl,2),nexttile(t_TCL_pha,2),"$x = f(-
    t)$");
37 CTFT_fig(t_2,x_ori,nexttile(t_TCL_ampl,3),nexttile(t_TCL_pha,3),"$x =
    f(2t)$");
38 CTFT_fig(t_3,x_ori,nexttile(t_TCL_ampl,4),nexttile(t_TCL_pha,4),"$x =
    f(t/2)$");
39 %
40 xlabel(t_TCL_ampl,"frequency (kHz)",FontSize=10,Interpreter="latex");
41 ylabel(t_TCL_ampl,"amplitude","FontSize",10,"Interpreter","latex");
42 [~,t_title_s] = title(t_TCL_ampl,"\bf CTFT (amplitude)","Guorui Wei
    516021910080",'Interpreter','latex');
43 set(t_title_s,'FontSize',8);
44 %
45 xlabel(t_TCL_pha,"frequency (kHz)",FontSize=10,Interpreter="latex");
46 ylabel(t_TCL_pha,"phase (rad)","FontSize",10,"Interpreter","latex");
47 [~,t_title_s] = title(t_TCL_pha,"\bf CTFT (phase)","Guorui Wei
    516021910080",'Interpreter','latex');
48 set(t_title_s,'FontSize',8);
49 %
50 %
    exportgraphics(t_TCL_ampl,"..\doc\fig\proj1\Fig_2a_CTFT_ampl_of_ordinal_
    data_and_its_time-
    domain_transformation.emf",'Resolution',800,'ContentType','auto','Background
    Color','none','Colorspace','rgb');
51 exportgraphics(t_TCL_ampl,"..\doc\fig\proj1\Fig_2a_CTFT_ampl_of_ordinal_
    data_and_its_time-
    domain_transformation.png",'Resolution',800,'ContentType','auto','Background
    Color','none','Colorspace','rgb');
52 %
    exportgraphics(t_TCL_pha,"..\doc\fig\proj1\Fig_2b_CTFT_phase_of_ordinal_
    data_and_its_time-
    domain_transformation.emf",'Resolution',800,'ContentType','auto','Background
    Color','none','Colorspace','rgb');
53 exportgraphics(t_TCL_pha,"..\doc\fig\proj1\Fig_2b_CTFT_phase_of_ordinal_
    data_and_its_time-
    domain_transformation.png",'Resolution',800,'ContentType','auto','Background
    Color','none','Colorspace','rgb');
54
```



```
55 % Inverse continuous-time Fourier transform, approximated by inverse DFT.
56
57 mag_spectrum = abs(CTFT_ori);
58 pha_spectrum = CTFT_ori./mag_spectrum;
59
60 [t_ori_IFT,x_ori_IFT] = ICTFT(omega_ori,CTFT_ori,t_ori(1));
61 [t_mag_IFT,x_mag_IFT] = ICTFT(omega_ori,mag_spectrum,t_ori(1));
62 [t_pha_IFT,x_pha_IFT] = ICTFT(omega_ori,pha_spectrum,t_ori(1));
63
64 %% Ideal filter
65
66 omega_c = 2500*2*pi; % [rad/s] cut-off frequency of ideal low-pass filter
    (LPF)
67 LPF_spec = omega_ori < omega_c | omega_ori > (2*pi*F_s - omega_c);
68 CTFT_filtered = CTFT_ori.*LPF_spec;
69 [t_filtered_IFT,x_filtered_IFT] = ICTFT(omega_ori,CTFT_filtered,t_ori(1));
70
71 %% Create figure.
72
73 %% Fig.1 orginal wave and its time-domain transformation
74
75 figure('Name',"Fig.1 orginal_wave_and_its_time-domain_transformation");
76 t_TCL = tiledlayout(2,2,"TileSpacing","tight","Padding","tight");
77 %
78 t_axes = nexttile(t_TCL,1);
79 plot(t_axes,t_ori,x_ori,'-',"DisplayName",'original waves');
80 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
    ','off','TickDir','out','XLimitMethod','tight');
81 title(t_axes,"$x = f(t)$",'Interpreter','latex');
82 %
    legend(t_axes,"Location",'best','Interpreter','latex',"Box","off",'FontSize'
    ,10);
83 %
84 t_axes = nexttile(t_TCL,2);
85 plot(t_axes,t_1,x_ori,'-',"DisplayName",'$x = f(-t)$');
86 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
    ','off','TickDir','out','YTickLabel',{},'XLimitMethod','tight');
87 title(t_axes,"$x = f(-t)$",'Interpreter','latex');
88 %
89 t_axes = nexttile(t_TCL,3);
90 plot(t_axes,t_2,x_ori,'-',"DisplayName",'$x = f(2t)$');
91 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
    ','off','TickDir','out','XLimitMethod','tight');
92 title(t_axes,"$x = f(2t)$",'Interpreter','latex');
```



```
93 %
94 t_axes = nexttile(t_TCL,4);
95 plot(t_axes,t_3,x_ori,'-', "DisplayName", '$x = f(t/2)$');
96 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
    ', 'off', 'TickDir', 'out', 'YTickLabel', {}, 'XLimitMethod', 'tight');
97 title(t_axes, "$x = f(t/2)$", 'Interpreter', 'latex');
98 %
99 xlabel(t_TCL, "time (seconds)", FontSize=10, Interpreter="latex");
100 ylabel(t_TCL, "signal", "FontSize", 10, "Interpreter", "latex");
101 [~,t_title_s] = title(t_TCL, "\bf Orginal waveform and its time-domain
    transformation", "Guorui Wei 516021910080", 'Interpreter', 'latex');
102 set(t_title_s, 'FontSize', 8);
103 %
104 %
    exportgraphics(t_TCL, "..\doc\fig\proj1\Fig_1_orginal_wave_and_its_time-
    domain_transformation.emf", 'Resolution', 800, 'ContentType', 'auto', 'Background
    Color', 'none', 'Colorspace', 'rgb');
105 exportgraphics(t_TCL, "..\doc\fig\proj1\Fig_1_orginal_wave_and_its_time-
    domain_transformation.png", 'Resolution', 800, 'ContentType', 'auto', 'Background
    Color', 'none', 'Colorspace', 'rgb');
106
107 %% Fig.3a inverse CTFT
108
109 figure('Name', "Fig.3a ICTFT: original signal");
110 t_TCL = tiledlayout(1,2, "TileSpacing", "tight", "Padding", "tight");
111 %
112 t_axes = nexttile(t_TCL,1);
113 plot(t_axes,t_ori_IFT,x_ori_IFT,'-', "DisplayName", 'ICTFT of original
    waves');
114 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
    ', 'off', 'TickDir', 'out', 'XLimitMethod', 'tight');
115 ylabel(t_axes, "signal", "FontSize", 10, "Interpreter", "latex");
116 title(t_axes, "$\widetilde{x}(t) := \mathcal{F}^{-
    1}[X(\omega)](t)$", 'Interpreter', 'latex');
117 % errors
118 x_err = (x_ori_IFT - x_ori)./abs(x_ori);
119 % x_err(abs(x_ori) < 0.01) = NaN;
120 t_axes = nexttile(t_TCL,2);
121 plot(t_axes,t_ori,x_err,'-', "DisplayName", 'ICTFT of original waves');
122 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
    ', 'off', 'TickDir', 'out', 'XLimitMethod', 'tight');
123 ylabel(t_axes, "relative error", "FontSize", 10, "Interpreter", "latex");
124 title(t_axes, "$(\widetilde{x}(t) - x(t))/|x(t)|$", 'Interpreter', 'latex');
125 %
```




```
126 xlabel(t_TCL,"time (seconds)",FontSize=10,Interpreter="latex");
127 [~,t_title_s] = title(t_TCL,"\bf Inverse Fourier Transform","Guorui Wei
128 516021910080",'Interpreter','latex');
129 set(t_title_s,'FontSize',8);
130 %
131 %
132 exportgraphics(t_TCL,"..\doc\fig\proj1\Fig_3a_ICTFT_original.emf",'Resol
133 ution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb')
134 ;
135 exportgraphics(t_TCL,"..\doc\fig\proj1\Fig_3a_ICTFT_original.png",'Resol
136 ution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb')
137 ;
138 %
139 %% Fig.3b inverse CTFT of magnitude spectrum
140
141 figure('Name',"Fig.3b ICTFT: magnitude spectrum");
142 t_TCL = tiledlayout(2,2,"TileSpacing","tight","Padding","tight");
143 %
144 t_axes = nexttile(t_TCL,1);
145 plot(t_axes,omega_ori/2/pi/1000,mag_spectrum,'-', ...
146      "DisplayName",' $|X(\omega)|$ ');
147 hold on
148 plot(t_axes,F_s/2/1000,0,'o', ...
149      "DisplayName",sprintf("Nyquist  $f_{\rm s}/2 = \%g$  kHz",F_s/2/1000));
150 hold off
151 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
152 ', 'off','TickDir','out','XLimitMethod','tight');
153 legend(t_axes,"Location",'best','Interpreter','latex',"Box","off",'FontSize'
154 ,10);
155 xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
156 ylabel(t_axes,"magnitude",FontSize=10,Interpreter,"latex");
157 title(t_axes," $|X(\omega)| :=
158 \|\mathcal{F}\{x(t)\}(\omega)\|$ ", 'Interpreter','latex');
159 %
160 t_axes = nexttile(t_TCL,2);
161 plot(t_axes,t_mag_IFT,real(x_mag_IFT),'-', "DisplayName",'ICTFT (real) of
162 magnitude spectrum');
163 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
164 ', 'off','TickDir','out','XLimitMethod','tight');
165 xlabel(t_axes," $t$  (seconds)",FontSize=10,Interpreter="latex");
166 ylabel(t_axes,"real part",FontSize=10,Interpreter,"latex");
167 title(t_axes," $\{\rm Re\} \left\{ \widetilde{x}(t) \right\} := \{\rm Re\}
168 \left\{ \mathcal{F}^{-1}[X(\omega)](t) \right\}$ ", 'Interpreter','latex');
169 %
```



```
158 t_axes = nexttile(t_TCL,3);
159 plot(t_axes,t_mag_IFT,imag(x_mag_IFT),'-',"DisplayName",'ICTFT (imag.) of
magnitude spectrum');
160 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
','off','TickDir','out','XLimitMethod','tight');
161 xlabel(t_axes,"$t$ (seconds)",FontSize=10,Interpreter="latex");
162 ylabel(t_axes,"imag. part","FontSize",10,"Interpreter","latex");
163 title(t_axes,"${\rm Im} \left\{ \widetilde{x}(t) \right\} := {\rm Im}
\left\{ \mathcal{F}^{-1}[X(\omega)](t) \right\}$", 'Interpreter', 'latex');
164 % errors
165 x_err = (x_mag_IFT - x_ori)./abs(x_ori);
166 x_err(abs(x_ori) < 0.01) = NaN;
167 t_axes = nexttile(t_TCL,4);
168 plot(t_axes,t_ori,x_err,'-',"DisplayName",'ICTFT of magnitude spectrum');
169 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
','off','TickDir','out','XLimitMethod','tight');
170 xlabel(t_axes,"$t$ (seconds)",FontSize=10,Interpreter="latex");
171 ylabel(t_axes,"relative error","FontSize",10,"Interpreter","latex");
172 title(t_axes,"$(\widetilde{x}(t) - x(t))/|x(t)|$", 'Interpreter', 'latex');
173 %
174 [~,t_title_s] = title(t_TCL,"\bf Inverse Fourier Transform of Magnitude
Spectrum","Guorui Wei 516021910080",'Interpreter','latex');
175 set(t_title_s,'FontSize',8);
176 %
177 %
exportgraphics(t_TCL,"..\doc\fig\proj1\Fig_3b_ICTFT_mag_spectrum.emf",'R
resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','r
gb');
178 exportgraphics(t_TCL,"..\doc\fig\proj1\Fig_3b_ICTFT_mag_spectrum.png",'R
resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','r
gb');
179
180 %% Fig.3c inverse CTFT of phase spectrum
181
182 figure('Name',"Fig.3c ICTFT: phase spectrum");
183 t_TCL = tiledlayout(2,2,"TileSpacing","tight","Padding","tight");
184 %
185 t_axes = nexttile(t_TCL,1);
186 plot(t_axes,omega_ori/2/pi/1000,angle(CTFT_ori),'-',"DisplayName",'phase
spectrum');
187 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
','off','TickDir','out','XLimitMethod','tight');
188 xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
189 ylabel(t_axes,"phase","FontSize",10,"Interpreter","latex");
```



```
190 title(t_axes,"${\rm arg} \left\{ X(\omega) \right\} := {\rm arg}
    \left\{ \mathcal{F}[x(t)](\omega) \right\}$", 'Interpreter', 'latex');
191 %
192 t_axes = nexttile(t_TCL,2);
193 plot(t_axes,t pha_IFT,real(x pha_IFT),'-', "DisplayName", 'ICTFT (real) of
    phase spectrum');
194 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
    ', 'off', 'TickDir', 'out', 'XLimitMethod', 'tight');
195 xlabel(t_axes, "$t$ (seconds)", FontSize=10, Interpreter="latex");
196 ylabel(t_axes, "real part", "FontSize", 10, "Interpreter", "latex");
197 title(t_axes, "$\widetilde{x}(t) := \mathcal{F}^{-1}
    \left[ X(\omega)/|X(\omega)| \right](t)$", 'Interpreter', 'latex');
198 %
199 t_axes = nexttile(t_TCL,3);
200 plot(t_axes,t pha_IFT,imag(x pha_IFT),'-', "DisplayName", 'ICTFT (imag.) of
    phase spectrum');
201 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
    ', 'off', 'TickDir', 'out', 'XLimitMethod', 'tight');
202 xlabel(t_axes, "$t$ (seconds)", FontSize=10, Interpreter="latex");
203 ylabel(t_axes, "imag. part", "FontSize", 10, "Interpreter", "latex");
204 title(t_axes, "$\widetilde{x}(t) := \mathcal{F}^{-1}
    \left[ X(\omega)/|X(\omega)| \right](t)$", 'Interpreter', 'latex');
205 % errors
206 x_err = (x pha_IFT - x_ori)./abs(x_ori);
207 x_err(abs(x_ori) < 0.01) = NaN;
208 t_axes = nexttile(t_TCL,4);
209 plot(t_axes,t ori,x_err,'-', "DisplayName", 'ICTFT of phase spectrum');
210 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
    ', 'off', 'TickDir', 'out', 'XLimitMethod', 'tight');
211 xlabel(t_axes, "$t$ (seconds)", FontSize=10, Interpreter="latex");
212 ylabel(t_axes, "relative error", "FontSize", 10, "Interpreter", "latex");
213 title(t_axes, "$(\widetilde{x}(t) - x(t))/|x(t)|$", 'Interpreter', 'latex');
214 %
215 [~,t_title_s] = title(t_TCL, "\bf Inverse Fourier Transform of Phase
    Spectrum", "Guorui Wei 516021910080", 'Interpreter', 'latex');
216 set(t_title_s, 'FontSize', 8);
217 %
218 %
    exportgraphics(t_TCL, "..\doc\fig\proj1\Fig_3c_ICTFT pha_spectrum.emf", 'R
    esolution', 800, 'ContentType', 'auto', 'BackgroundColor', 'none', 'Colorspace', 'r
    gb');
219 exportgraphics(t_TCL, "..\doc\fig\proj1\Fig_3c_ICTFT pha_spectrum.png", 'R
    esolution', 800, 'ContentType', 'auto', 'BackgroundColor', 'none', 'Colorspace', 'r
    gb');
```



```
220
221 %% Fig.4 waveform after passing an ideal LPF
222
223 figure('Name',"Fig.4 ideal-LPF-filtered waveform");
224 t_TCL = tiledlayout(2,2,"TileSpacing","tight","Padding","tight");
225 %
226 t_axes = nexttile(t_TCL,1);
227 plot(t_axes,omega_ori/2/pi/1000,mag_spectrum,'-', ...
228      "DisplayName",' $|X(\omega)|$ ');
229 hold on
230 plot(t_axes,[omega_c/2/pi/1000,omega_c/2/pi/1000],[0,max(mag_spectrum)],'--
231      ', ...
232      "DisplayName",sprintf("cut-off  $f_{\text{c}}$ 
233      = %g$ kHz",omega_c/2/pi/1000));
234 plot(t_axes,F_s/2/1000,0,'o', ...
235      "DisplayName",sprintf("Nyquist  $f_{\text{s}}/2 = %g$ kHz",F_s/2/1000));
236 hold off
237 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
238      ','off','TickDir','out','XLimitMethod','tight');
239 legend(t_axes,"Location",'best','Interpreter','latex',"Box","off",'FontSize'
240      ',10);
241 xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
242 ylabel(t_axes,"magnitude","FontSize",10,"Interpreter","latex");
243 title(t_axes," $|X(\omega)| :=
244      |\mathcal{F}\{x(t)\}(\omega)|$ ", 'Interpreter','latex');
245 %
246 t_axes = nexttile(t_TCL,2);
247 plot(t_axes,omega_ori/2/pi/1000,LPF_spec,'-', ...
248      "DisplayName",' $H(f)$ ');
249 hold on
250 plot(t_axes,omega_c/2/pi/1000,0,'^', ...
251      "DisplayName",sprintf("cut-off  $f_{\text{c}}$ 
252      = %g$ kHz",omega_c/2/pi/1000));
253 plot(t_axes,F_s/2/1000,0,'o', ...
254      "DisplayName",sprintf("Nyquist  $f_{\text{s}}/2 = %g$ kHz",F_s/2/1000));
255 hold off
256 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
257      ','off','TickDir','out','XLimitMethod','tight');
258 legend(t_axes,"Location",'best','Interpreter','latex',"Box","off",'FontSize'
259      ',10);
260 xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
261 ylabel(t_axes,"magnitude","FontSize",10,"Interpreter","latex");
262 title(t_axes,sprintf("ideal LPF:  $H(\omega) :=
263      \mathcal{F}\{h(t)\}(\omega)$ ", 'Interpreter','latex');$$ 
```



```
255 %
256 t_axes = nexttile(t_TCL,3);
257 plot(t_axes,t_filtered_IFT,x_filtered_IFT,'-', "DisplayName", 'LPF-filtered
waveform');
258 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
', 'off', 'TickDir', 'out', 'XLimitMethod', 'tight');
259 xlabel(t_axes, "$t$ (seconds)", FontSize=10, Interpreter="latex");
260 ylabel(t_axes, "amplitude", "FontSize", 10, "Interpreter", "latex");
261 title(t_axes, "$\widetilde{x}(t) := \mathcal{F}^{-1}[X(\omega)H(\omega)](t)$", 'Interpreter', 'latex');
262 % errors
263 x_err = (x_filtered_IFT - x_ori)./abs(x_ori);
264 x_err(abs(x_ori) < 0.01) = NaN;
265 t_axes = nexttile(t_TCL,4);
266 plot(t_axes,t_ori,x_err,'-', "DisplayName", 'LPF-filtered waveform');
267 set(t_axes, "YDir", 'normal', "TickLabelInterpreter", 'latex', "FontSize", 10, 'Box
', 'off', 'TickDir', 'out', 'XLimitMethod', 'tight');
268 xlabel(t_axes, "$t$ (seconds)", FontSize=10, Interpreter="latex");
269 ylabel(t_axes, "relative error", "FontSize", 10, "Interpreter", "latex");
270 title(t_axes, "$(\widetilde{x}(t) - x(t))/|x(t)|$", 'Interpreter', 'latex');
271 %
272 [~,t_title_s] = title(t_TCL, "\bf ideal-LPF-filtered waveform", "Guorui Wei
516021910080", 'Interpreter', 'latex');
273 set(t_title_s, 'FontSize', 8);
274 %
275 %
exportgraphics(t_TCL, "..\doc\fig\proj1\Fig_4_LPF_filtered_signal.emf", 'R
resolution', 800, 'ContentType', 'auto', 'BackgroundColor', 'none', 'Colorspace', 'r
gb');
276 exportgraphics(t_TCL, "..\doc\fig\proj1\Fig_4_LPF_filtered_signal.png", 'R
resolution', 800, 'ContentType', 'auto', 'BackgroundColor', 'none', 'Colorspace', 'r
gb');
277
278 %% local functions
279
280 %% Initialize environment
281
282 function [] = init_env()
283 % Initialize environment
284 %
285 % set up project directory
286 if ~isfolder("../doc/fig/proj1")
287     mkdir ../doc/fig/proj1
288 end
```



```
289 % configure searching path
290 mfile_fullpath = mfilename('fullpath'); % the full path and name of the
      file in which the call occurs, not including the filename extension.
291 mfile_fullpath_without_fname = mfile_fullpath(1:end-
      strlen(mfilename));
292 addpath(genpath(mfile_fullpath_without_fname + "../data"), ...
293         genpath(mfile_fullpath_without_fname + "../inc")); % adds the
      specified folders to the top of the search path for the current MATLAB®
      session.
294
295 return;
296 end
297
298 %% CTFT by DFT
299
300 function [omega,X] = CTFT(t,x)
301 % Continuous-time Fourier transform, approximated by DFT.
302 % CAVEAT: MUST be sampled at equal intervals!
303 arguments
304     t
305     x
306 end
307
308 if(~iscolumn(t))
309     t = t.';
310 end
311 if(~iscolumn(x))
312     x = x.';
313 end
314
315 [t,I] = sort(t);
316 x = x(I);
317
318 T_s = t(2) - t(1);
319 N = length(t);
320
321 omega = (0:N-1).'/N*2*pi/T_s;
322 X = T_s * fft(x) .* exp(-omega*t(1)*1i);
323 end
324
325 %% Tool function for Fig.2 (CTFT)
326
327 function [omega,X] = CTFT_fig(t,x,axes_amp,axes_pha,str_title)
328 % H1
```



```
329 % details
330     arguments
331         t
332         x
333         axes_amp
334         axes pha
335         str_title = "$x = f(t)$"
336     end
337
338     [omega,X] = CTFT(t,x);
339 %
340     N_half = ceil(length(omega)/2);
341     plot(axes_amp,omega(1:N_half)/2/pi/1000,abs(X(1:N_half)),'-', ...
342         "DisplayName",str_title);
343     set(axes_amp,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",1
344         0,'Box','off','TickDir','out','XLimitMethod','tight');
345     title(axes_amp,str_title,'Interpreter','latex');
346 %
347     plot(axes pha,omega(1:N_half)/2/pi/1000,angle(X(1:N_half)),'-', ...
348         "DisplayName",str_title);
349     set(axes pha,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",1
350         0,'Box','off','TickDir','out','XLimitMethod','tight');
351     title(axes pha,str_title,'Interpreter','latex');
352 end
353
354 %% ICTFT by IDFT
355
356 function [t,x] = ICTFT(omega,X,t0)
357 % inverse continuous-time Fourier transform, approximated by IDFT.
358 % CAVEAT: MUST be sampled at equal intervals!
359
360     arguments
361         omega
362         X
363         t0 = 0;
364     end
365
366     if(~iscolumn(omega))
367         omega = omega.';
368     end
369
370     if(~iscolumn(X))
371         X = X.';
372     end
373
374     [omega,I] = sort(omega);
```



```
371 X = X(I);
372
373 N = length(omega);
374 omega_s = (omega(2) - omega(1))*N;
375 T_s = 2*pi/omega_s;
376
377 t = t0 + (0:N-1).*T_s;
378 x = ifft(X.*exp(1i*omega*t0))/T_s;
379 end
380
```

A.2 proj2_2.m

```
1 %% proj2_2.m
2 % Description: MATLAB code for course project (ICE2301, 2022 Spring)
3 % Author: Guorui Wei (危国锐) (313017602@qq.com; weiguorui@sjtu.edu.cn)
4 % Student ID: 516021910080
5 % Created: 2022-05-26
6 % Last modified: 2022-05-28
7
8 %% Initialize project
9
10 clc; clear; close all
11 init_env();
12
13 %% Create original signal.
14
15 F_ori = 10000./(2.^(0:2)); % frequency of components of original signal
    [MUST be geometric series!]
16 phi_ori = [pi/6,pi/2,5*pi/6]; % initial phase of components of original
    signal
17 F_s = 64*max(F_ori); % sampling frequency
18 tau_s = 64/min(F_ori); % sampling duration
19
20 t_o = linspace(0,tau_s-1/F_s,F_s*tau_s).'; % time-interval
21 x_o = x_ori(t_o,F_ori,phi_ori); % original signal
22 [omega_ori,CTFT_ori] = CTFT(t_o,x_o); % CTFT of original signal
23 cycle_ori = 1/min(F_ori); % cycle of original signal
24
25 %% Create figure.
26
27 figure('Name',"Fig.0 original signal");
28 t_TCL = tiledlayout(2,2,"TileSpacing","tight","Padding","tight");
29 % CTFT of original signal (mag.)
```




```
30 t_axes = nexttile(t_TCL,1);
31 plot(t_axes,omega_ori/2/pi/1000,abs(CTFT_ori),'-', ...
32      "DisplayName"," $X_{\text{rm o}}(f)$ ");
33 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
34     ','off','TickDir','out');
35 xlim(t_axes,[0,2*max(F_ori)/1000]);
36 xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
37 ylabel(t_axes,"magnitude",FontSize=10,Interpreter,"latex");
38 title(t_axes," $X_{\text{rm o}}(\omega) := \mathcal{F}[x_{\text{rm
39     o}}(t)](\omega)$ ", 'Interpreter','latex');
40 % CTFT of original signal (pha.)
41 t_axes = nexttile(t_TCL,2);
42 plot(t_axes,omega_ori/2/pi/1000,angle(CTFT_ori),'-', ...
43      "DisplayName"," $\arg\{X_{\text{rm o}}(f)\}$ ");
44 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
45     ','off','TickDir','out','XLim',[0,2*max(F_ori)/1000]);
46 xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
47 ylabel(t_axes,"phase (rad)",FontSize=10,Interpreter,"latex");
48 title(t_axes," $X_{\text{rm o}}(\omega) := \mathcal{F}[x_{\text{rm
49     o}}(t)](\omega)$ ", 'Interpreter','latex');
50 % original signal
51 t_axes = nexttile(t_TCL,3,[1 2]);
52 plot(t_axes,t_o(1:2*F_s*cycle_ori+1)*1e6,x_o(1:2*F_s*cycle_ori+1),'-', ...
53      "DisplayName"," $x_{\text{rm o}}(t)$ ");
54 set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box
55     ','off','TickDir','out','XLimitMethod','tight');
56 xlabel(t_axes," $t$  ( $\mu\text{s}$ )",FontSize=10,Interpreter="latex");
57 ylabel(t_axes,"amplitude",FontSize=10,Interpreter,"latex");
58 title(t_axes,sprintf("\bf original signal  $x_{\text{rm o}}(t)$  (periodic,  $T
59     = \%g$   $\mu\text{s}$ )",1e6/min(F_ori)), 'Interpreter','latex');
60 %
61 [~,t_title_s] = title(t_TCL,sprintf("\bf original signal
62     (periodic)"),'Guorui Wei 516021910080', 'Interpreter','latex');
63 set(t_title_s,'FontSize',8);
64 %
65 exportgraphics(t_TCL,sprintf("../doc\\fig\\proj2_2\\Fig_0_original_signal.e
66     mf"),'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorsp
67     ace','rgb');
68 exportgraphics(t_TCL,sprintf("../doc\\fig\\proj2_2\\Fig_0_original_signal.p
69     ng"),'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorsp
70     ace','rgb');
71
72 %% Recover signal by ideal band-limited interpolation.
```



```
63 %% parameters
64
65 F_s = 44e3./(2.^(0:4)); % sampling frequency
66 t_s0 = 0; % sampling start time
67 tau_s = cycle_ori*128; % sampling duration
68 F_r = 32*max(F_ori); % recovering frequency
69 t_r0 = t_s0; % recovering start time
70 tau_r = tau_s; % recovering duration
71
72 for i = 1:length(F_s)
73     ibli_fig(F_s(i),t_s0,tau_s,F_r,t_r0,tau_r,F_ori,phi_ori);
74 end
75
76 %% local functions
77
78 %% Initialize environment
79
80 function [] = init_env()
81 % Initialize environment
82 %
83 % set up project directory
84 if ~isfolder("../doc/fig/proj2_2")
85     mkdir ../doc/fig/proj2_2
86 end
87 % configure searching path
88 mfile_fullpath = mfilename('fullpath'); % the full path and name of the
file in which the call occurs, not including the filename extension.
89 mfile_fullpath_without_fname = mfile_fullpath(1:end-
strlength(mfilename));
90 addpath(genpath(mfile_fullpath_without_fname + "../data"), ...
91         genpath(mfile_fullpath_without_fname + "../inc")); % adds the
specified folders to the top of the search path for the current MATLAB®
session.
92 return;
93 end
94
95 %% Create original signal.
96
97 function [x] = x_ori(t,F_ori,phi_ori)
98 %x_ori - original signal
99 %
100 % Syntax: [x] = x_ori(t,F_ori,phi_ori)
101 %
102 % create original signal.
```



```
103     arguments
104         t = 0;
105         F_ori = 1;
106         phi_ori = 0;
107     end
108
109     x = zeros(size(t));
110     for i = 1:length(F_ori)
111         x = x + sin(2*pi*F_ori(i)*t + phi_ori(i));
112     end
113 end
114
115 %% ideal band-limited interpolation
116
117 function x_r = ibli(t_n,x_n,t_r)
118 %ibli - ideal band-limited interpolation
119 %
120 % Syntax: x_r = ibli(t_n,x_n,t_r)
121 %
122 % ideal band-limited interpolation
123     arguments
124         t_n % sample time
125         x_n % sample value
126         t_r % recover value
127     end
128
129     if(~iscolumn(x_n))
130         x_n = x_n.';
131     end
132
133     T_s = abs(t_n(2) - t_n(1));
134     [grid_t_r,grid_t_n] = ndgrid(t_r,t_n);
135     x_r = sinc((grid_t_r - grid_t_n)/T_s)*x_n;
136 end
137
138 %% CTFT by DFT
139
140 function [omega,X] = CTFT(t,x)
141 % Continuous-time Fourier transform, approximated by DFT.
142 % CAVEAT: MUST be sampled at equal intervals!
143     arguments
144         t
145         x
```



```
146     end
147
148     if(~iscolumn(t))
149         t = t.';
150     end
151     if(~iscolumn(x))
152         x = x.';
153     end
154
155     [t,I] = sort(t);
156     x = x(I);
157
158     T_s = t(2) - t(1);
159     N = length(t);
160
161     omega = (0:N-1).'/N*2*pi/T_s;
162     X = T_s * fft(x) .* exp(-omega*t(1)*1i);
163 end
164
165 % Recover signal, by ideal band-limited interpolation
166
167 function [] = ibli_fig(F_s,t_s0,tau_s,F_r,t_r0,tau_r,F_ori,phi_ori)
168     arguments
169         F_s = 22000;           % sampling frequency
170         t_s0 = 0;             % sampling start time
171         tau_s = 1/F_s;        % sampling duration
172         F_r = F_s;           % recovering frequency
173         t_r0 = t_s0;         % recovering start time
174         tau_r = tau_s;        % recovering duration
175         F_ori = [10000,5000]; % frequency of components of original signal
176         phi_ori = [0,0];     % initial phase of components of original
177     end
178
179     cycle_ori = 1/min(F_ori); % cycle of original
180     signal
181     t_n = linspace(t_s0,t_s0+tau_s,F_s*tau_s+1).'; % sampling time
182     x_n = x_ori(t_n,F_ori,phi_ori); % sampling values
183     t_r = linspace(t_r0,t_r0+tau_r,F_r*tau_r+1).'; % recovering time
184     x_r = ibli(t_n,x_n,t_r); % recovering values
185     x_r_ori = x_ori(t_r,F_ori,phi_ori); % original values at
186     recovering time
```



```
185     [omega_r,CTFT_r] = CTFT(t_r,x_r);           % CTFT of recovering
signal
186     % recovering error
187     err_r = (x_r - x_r_ori)./abs(x_r_ori);
188     err_r(abs(x_r_ori) < 0.05) = NaN;
189
190     %% Create figure.
191
192     figure('Name',"Fig.1 ideal band-limited interpolation");
193     t_TCL = tiledlayout(2,2,"TileSpacing","tight","Padding","tight");
194     % CTFT of recovering signal (mag.)
195     t_axes = nexttile(t_TCL,2);
196     plot(t_axes,omega_r/2/pi/1000,abs(CTFT_r),'-', ...
197          "DisplayName"," $|X_{\rm r}(f)|$ ");
198     hold on
199     plot(t_axes,F_s/2/1000,0,'o', ...
200          "DisplayName",sprintf("Nyquist  $f_{\rm s}/2$ 
= %g$ kHz",F_s/2/1000));
201     hold off
202     set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,
'Box','off','TickDir','out');
203     xlim(t_axes,[0,2*max(F_ori)/1000]);
204     legend(t_axes,"Location",'best','Interpreter','latex',"Box","off",'FontS
ize',10);
205     xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
206     ylabel(t_axes,"magnitude","FontSize",10,"Interpreter","latex");
207     title(t_axes,sprintf(" $X_{\rm r}(\omega) := \mathcal{F}[x_{\rm r}
r(t)(\omega), $N_{\rm r} = %g$ ",length(t_r))','Interpreter','latex');
208     % CTFT of recovering signal (pha.)
209     t_axes = nexttile(t_TCL,4);
210     plot(t_axes,omega_r/2/pi/1000,angle(CTFT_r),'-', ...
211          "DisplayName"," $\arg\{X_{\rm r}(f)\}$ ");
212     set(t_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,
'Box','off','TickDir','out');
213     xlim(t_axes,[0,2*max(F_ori)/1000]);
214 %     legend(t_axes,"Location",'best','Interpreter','latex',"Box","off",'Fon
tSize',10);
215     xlabel(t_axes,"frequency (kHz)",FontSize=10,Interpreter="latex");
216     ylabel(t_axes,"phase (rad)","FontSize",10,"Interpreter","latex");
217     title(t_axes," $X_{\rm r}(\omega) := \mathcal{F}[x_{\rm r}
r(t)(\omega)$ ", 'Interpreter','latex');
218     % recovering signal
219     TF_t_r_range = t_r >= t_s0-cycle_ori & t_r <= t_s0+tau_s+cycle_ori; %
only show recovering signal at sampling time-interval$ 
```



```
220     t_axes = nexttile(t_TCL,1);
221     plot(t_axes,t_r(TF_t_r_range)*1e3,x_r(TF_t_r_range),'-
', "DisplayName", "$x_{\rm r}(t)$");
222     set(t_axes,"YDir","normal","TickLabelInterpreter","latex","FontSize",10,
'Box','off','TickDir','out','XLimitMethod','tight');
223     xlabel(t_axes,"$t$ $\rm (ms)$",FontSize=10,Interpreter="latex");
224     ylabel(t_axes,"amplitude","FontSize",10,"Interpreter","latex");
225     title(t_axes,"\bf recovering signal $x_{\rm
r}(t)$",'Interpreter','latex');
226     % recovering errors
227     t_axes = nexttile(t_TCL,3);
228     plot(t_axes,t_r*1e3,err_r,'-', "DisplayName", 'LPF-filtered waveform');
229     set(t_axes,"YDir","normal","TickLabelInterpreter","latex","FontSize",10,
'Box','off','TickDir','out','XLimitMethod','tight');
230     xlabel(t_axes,"$t$ $\rm (ms)$",FontSize=10,Interpreter="latex");
231     ylabel(t_axes,"relative error","FontSize",10,"Interpreter","latex");
232     title(t_axes,"$e_{\rm r}(t) := (x_{\rm r}(t) -
x(t))/|x(t)|$", 'Interpreter','latex');
233     %
234     [~,t_title_s] = title(t_TCL,sprintf("\bf ideal band-limited
interpolation} ($f_{\rm s} = %g$ kHz, $N_{\rm s}
= %g$)",F_s/1000,length(t_n)),"Guorui Wei
516021910080",'Interpreter','latex');
235     set(t_title_s,'FontSize',8);
236     %
237     %
exportgraphics(t_TCL,sprintf("../doc\\fig\\proj2_2\\Fig_%s_kHz_ideal_band_l
imited_interp.emf",num2str(F_s/1000)),'Resolution',800,'ContentType','auto',
'BackgroundColor','none','Colorspace','rgb');
238     exportgraphics(t_TCL,sprintf("../doc\\fig\\proj2_2\\Fig_%s_kHz_ideal_ba
nd_limited_interp.png",num2str(F_s/1000)),'Resolution',800,'ContentType','au
to','BackgroundColor','none','Colorspace','rgb');
239 end
240
```