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变换可视作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](#_Toc104624824)

[Abstract ii](#_Toc104624825)

[目 录 iii](#_Toc104624826)

[1 项目要求 1](#_Toc104624827)

[2 预备知识 2](#_Toc104624828)

[2.1 离散Fourier变换（DFT） 2](#_Toc104624829)

[2.2 离散时间Fourier变换（DTFT） 2](#_Toc104624830)

[2.3 Fourier级数（FS） 3](#_Toc104624831)

[2.4 连续时间Fourier变换（CTFT） 3](#_Toc104624832)

[2.5 DFT与CTFT的关系：从数值积分的视角 4](#_Toc104624833)

[3 项目1 6](#_Toc104624834)

[3.1 项目简介 6](#_Toc104624835)

[3.2 数据和方法 6](#_Toc104624836)

[3.2.1 音频信号 6](#_Toc104624837)

[3.2.2 CTFT数值算法 6](#_Toc104624838)

[3.2.3 数值CTFT的性质 6](#_Toc104624839)

[3.3 结果和讨论 9](#_Toc104624840)

[3.4 项目小结 15](#_Toc104624841)

[4 项目2：语音采样器（Speech Sampler） 16](#_Toc104624842)

[4.1 项目简介 16](#_Toc104624843)

[4.2 数据和方法 16](#_Toc104624844)

[4.2.1 连续时间信号 16](#_Toc104624845)

[4.2.2 冲激串采样与理想带限内插 17](#_Toc104624846)

[4.2.3 基于理想带限内插的信号重建 19](#_Toc104624847)

[4.2.4 信号重建质量的评价 19](#_Toc104624848)

[4.3 结果和讨论 20](#_Toc104624849)

[4.3.1 重建的信号 20](#_Toc104624850)

[4.3.2 讨论 20](#_Toc104624851)

[4.4 项目小结 24](#_Toc104624852)

[References 25](#_Toc104624853)

[Appendix A MATLAB源代码 26](#_Toc104624854)

[A.1 proj1.m 26](#_Toc104624855)

[A.2 proj2\_2.m 37](#_Toc104624856)

# 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, 画出该信号的波形.
2. 生成 , 并画出它们的波形.
3. 分别计算 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. 将的傅里叶变换记为，画出幅度谱的傅里叶反变换的波形，画出相位谱的傅里叶反变换的波形. 将它们与原始信号进行比较.
5. 对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.

# 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（）.

## 离散Fourier变换（DFT）

在一定的条件下，以正整数为最小正周期的离散时间信号可表为个以为周期的复指数信号

的线性组合：

若将视作在函数内积空间

上的最佳平方逼近的坐标，则满足法方程

其中

从而

定义

为离散时间序列的点离散Fourier变换（Discrete Fourier transform, DFT），则

为相应的点离散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.

## 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）.

设是一个非周期的离散时间信号. 定义以偶数为最小正周期的离散时间信号

有

由IDFT（），有

定义

为离散时间序列的离散时间Fourier变换（Discrete-time Fourier transform, DTFT），则

为相应的离散时间Fourier逆变换. 这样，非周期的离散时间信号被表为复指数信号的“线性组合”.

注意到是的周期. 可见，非周期、离散时间信号的频谱是连续的、周期的.

## Fourier级数（FS）

在一定的条件下，以为最小正周期的连续时间信号可表为可列个以为周期的复指数信号

的线性组合：

称（）为周期的连续时间信号的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.

## 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）.

设是一个非周期的连续时间信号. 定义以为最小正周期的连续时间信号

有

由周期的连续时间信号的Fourier级数（），有

定义

为连续时间信号的连续时间Fourier变换（Continuous-time Fourier transform, CTFT），则

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被表为复指数信号的“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.

## 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.

设有连续时间信号满足

这仅在有限时间上不恒为零，故称其为时域有限的. 记的CTFT为

设在计算机中存储了在时刻的采样值 记为采样频率，则. 记离散时间序列

的点DFT（）为

使用采样时刻上的值，定积分（）的矩形近似为

比较（）和（），得的CTFT与其采样序列的点DFT的关系为（在矩形积分公式近似下）

其中为采样圆频率. 记上式写成

式（）右端可作为的CTFT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（）只能得到在上的一些近似值（若取则得到上的），这符合Nyquist采样定理.

注2.2 事实上，不必限制（）和（）中的取值. 因为具有周期所以CTFT数值解具有周期.

# 项目1

## 项目简介

选择一段音频信号 (音乐等), 将其记为, 画出该信号的波形.

生成 , 并画出它们的波形.

分别计算 的傅里叶变换，画出它们的频谱，并对它们进行比较和分析.

将的傅里叶变换记为，画出幅度谱的傅里叶反变换的波形，画出相位谱的傅里叶反变换的波形. 将它们与原始信号进行比较.

对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.

## 数据和方法

### 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/)工具[[1]](#footnote-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.

### CTFT数值算法

设在计算机中存储了连续时间信号在时刻的样值

采样间隔下面建立数值求解的CTFT的算法.

假定[[2]](#footnote-2). 的CTFT可按（）求得数值结果：

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]](#footnote-3)）计算.

至此，数值求解的CTFT的算法已建立，现重述如下.

算法3.1 (CTFT by DFT) **输入**：数组 要求单调递增，且等间隔.

**步1** 决定 令 为数组的首元素，为的公差.

**步2** 按（）计算

### 数值CTFT的性质

记是用算法3.1求出的的数值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.

设是由的样值

用算法3.1求得的，简记为

其中

#### 反变换

由离散Fourier逆变换（），有

上式右端可视作连续时间Fourier逆变换（）

的矩形积分公式的近似，其中是的CTFT解析解（）

式（）可写成

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]](#footnote-4)）计算.

算法3.2 (ICTFT by IDFT) **输入**：初始时刻数组 要求单调递增，且等间隔.

**步1** 决定 令为的公差，则

**步2** 按（）计算

#### 尺度变换性质

考虑信号. 若则它的数值CTFT将由其样值

决定. 根据算法3.1，

若 则它的CTFT数值解将由其样值

决定. 根据算法3.1，此时

式（）（）可统一写成

其中

式（）与CTFT解析解的尺度变换性质相同.

#### 时移性质

考虑信号则它的数值CTFT数值将由其样值

决定. 根据算法3.1，

上式简记为

其中

式（）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.

#### 奇偶虚实性

考虑的共轭信号用算法3.1得到的的CTFT数值解

上式简记为

其中

式（）与CTFT解析解的相应性质相同.

特别地，若为实信号，则是共轭对称的：即

反之，若是共轭对称的，则是实的.

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.

## 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.

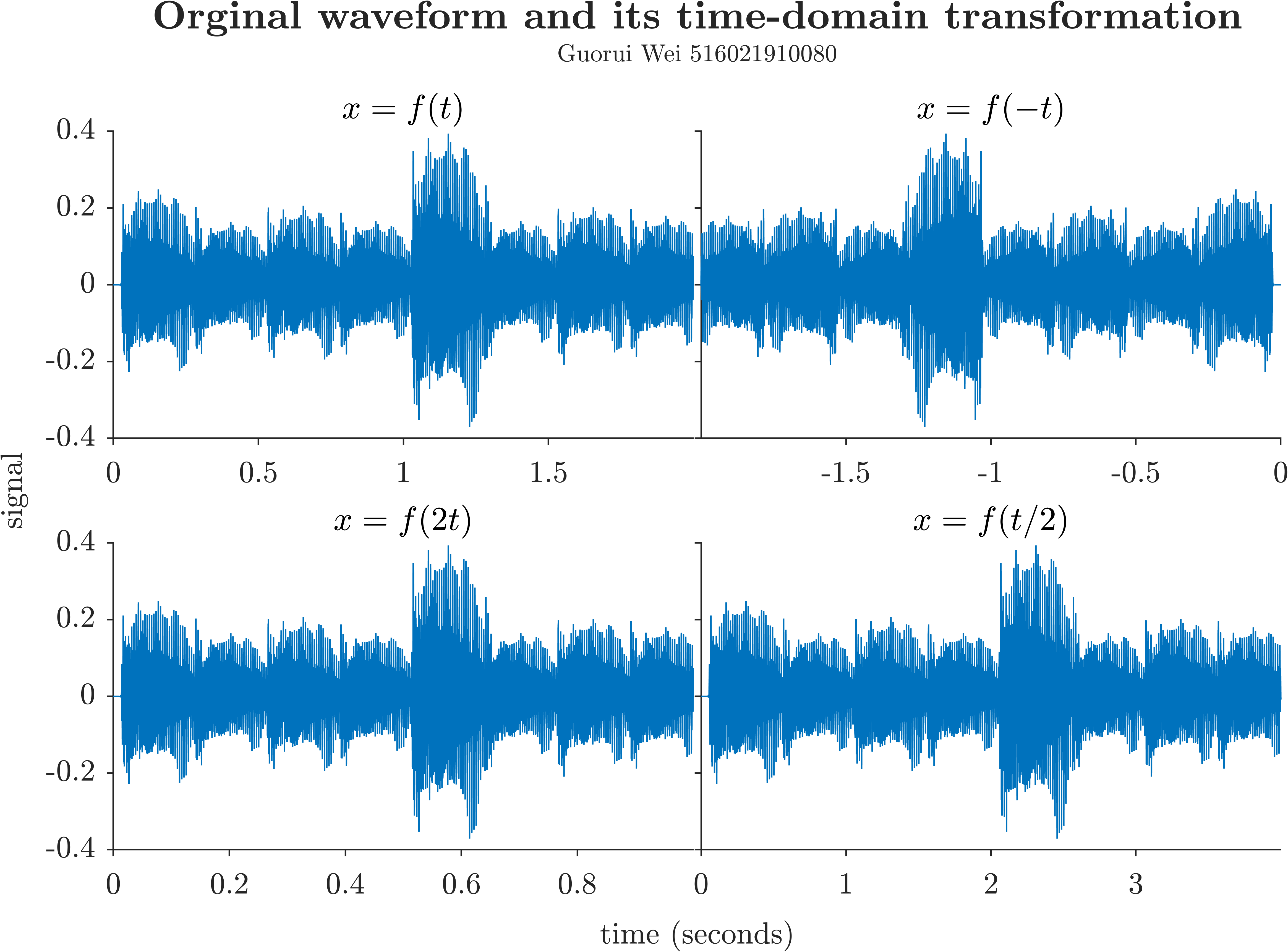
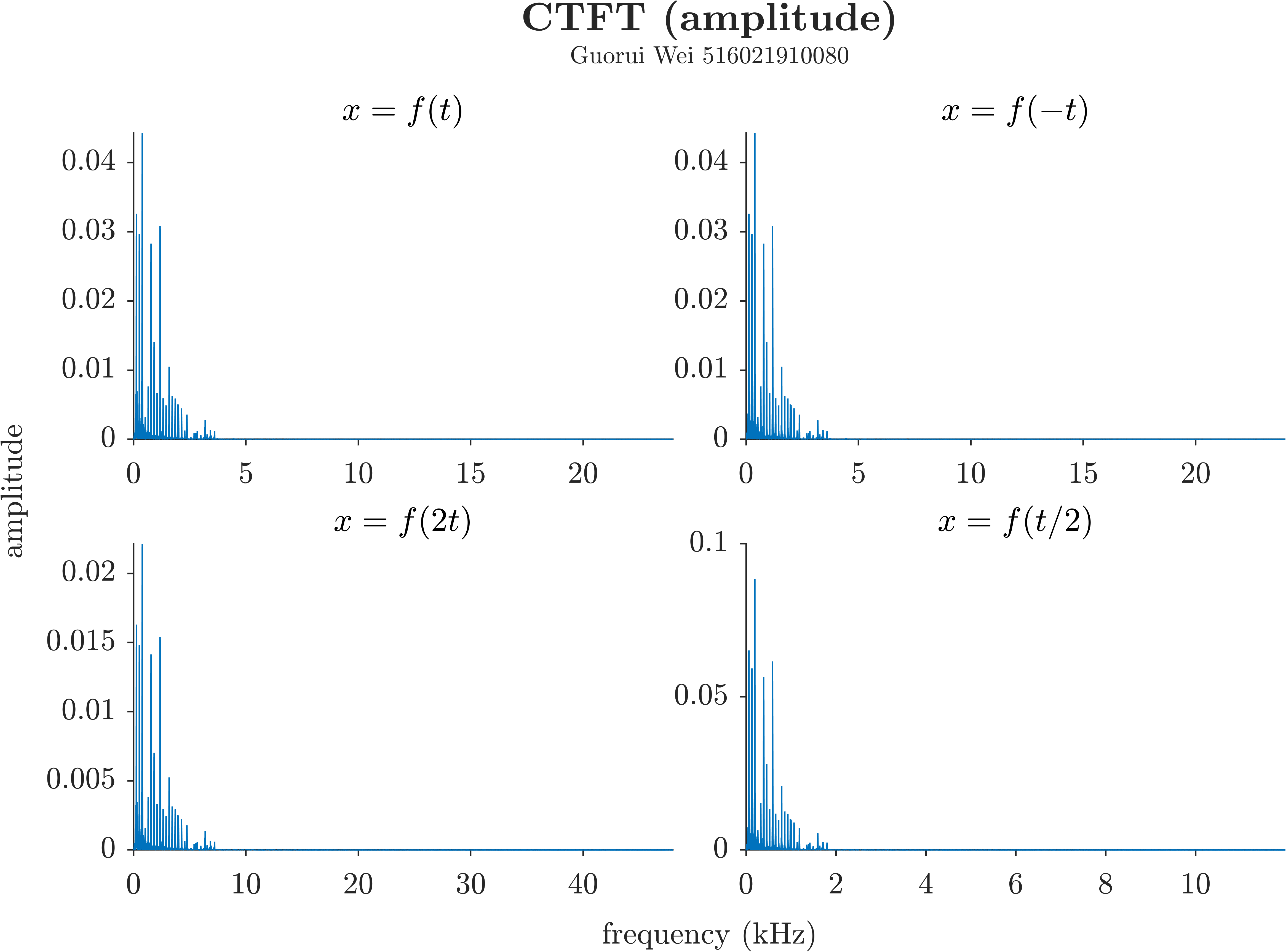
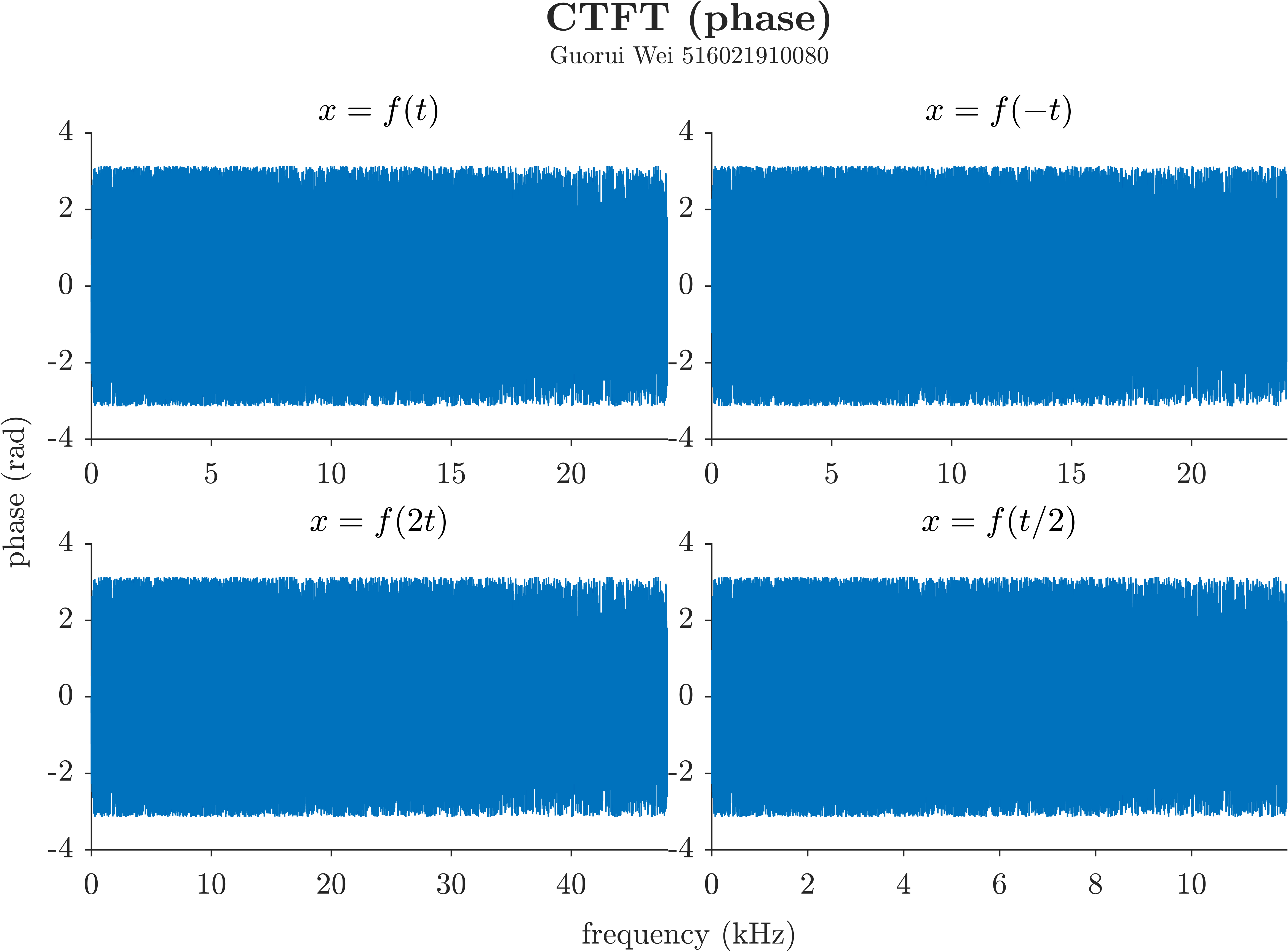


图3.1 原始音频信号，及其时域尺度变换. 注意横坐标轴刻度的差异.



（a）



（b）

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.

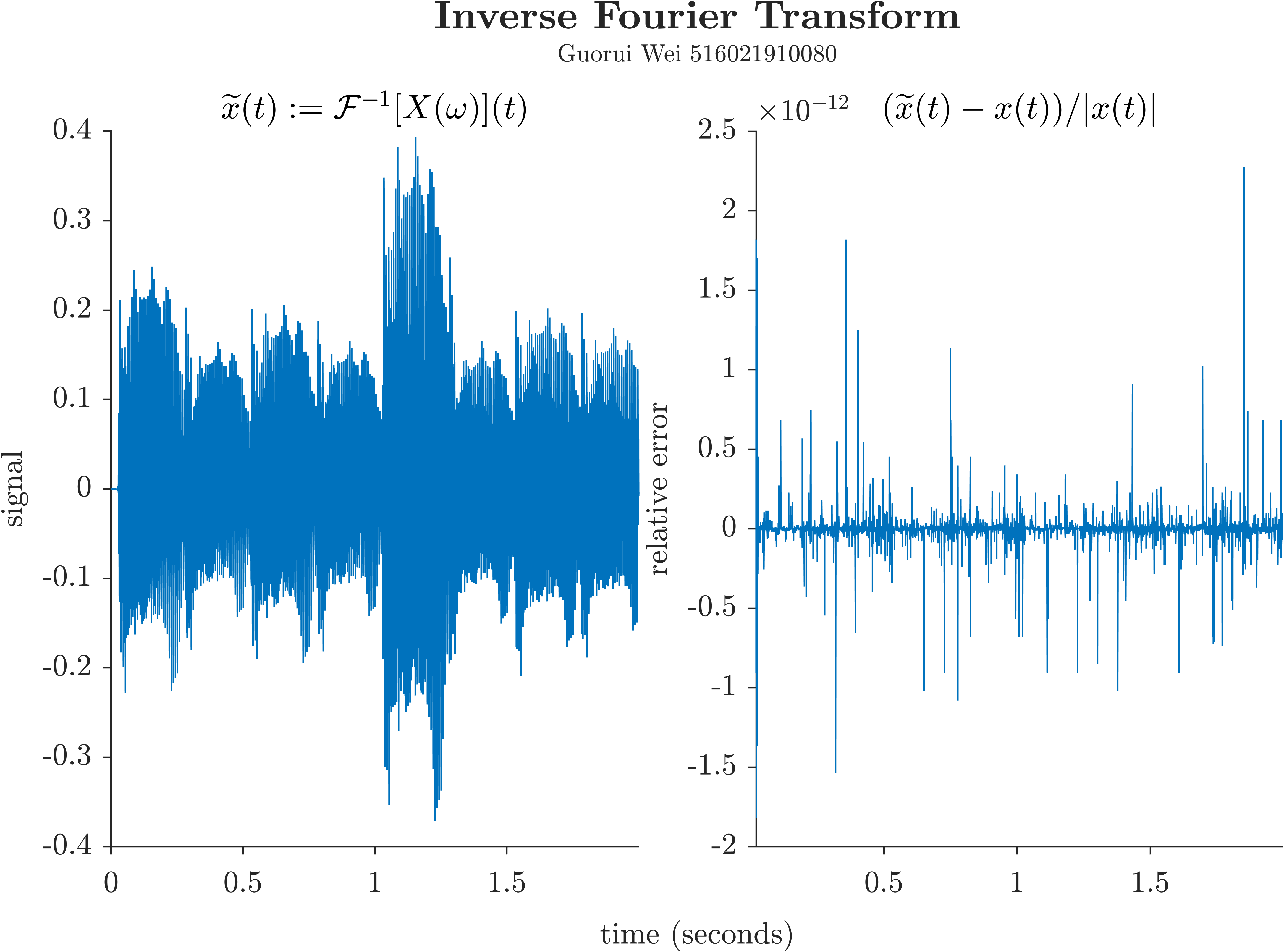


图3.3 对原始信号的数值连续时间Fourier变换（CTFT）用算法3.2求数值连续时间Fourier逆变换（inverse continuous-time Fourier transform, ICTFT）的结果. 求出的ICTFT是实的（与命题3.4一致），波形示于左子图. 求出的数值ICTFT对原始信号的相对误差示于右子图，可见这数值ICTFT与原始信号的差别基本上只是数值噪声，说明用算法3.1求数值CTFT与用算法3.2求数值ICTFT是互逆的.

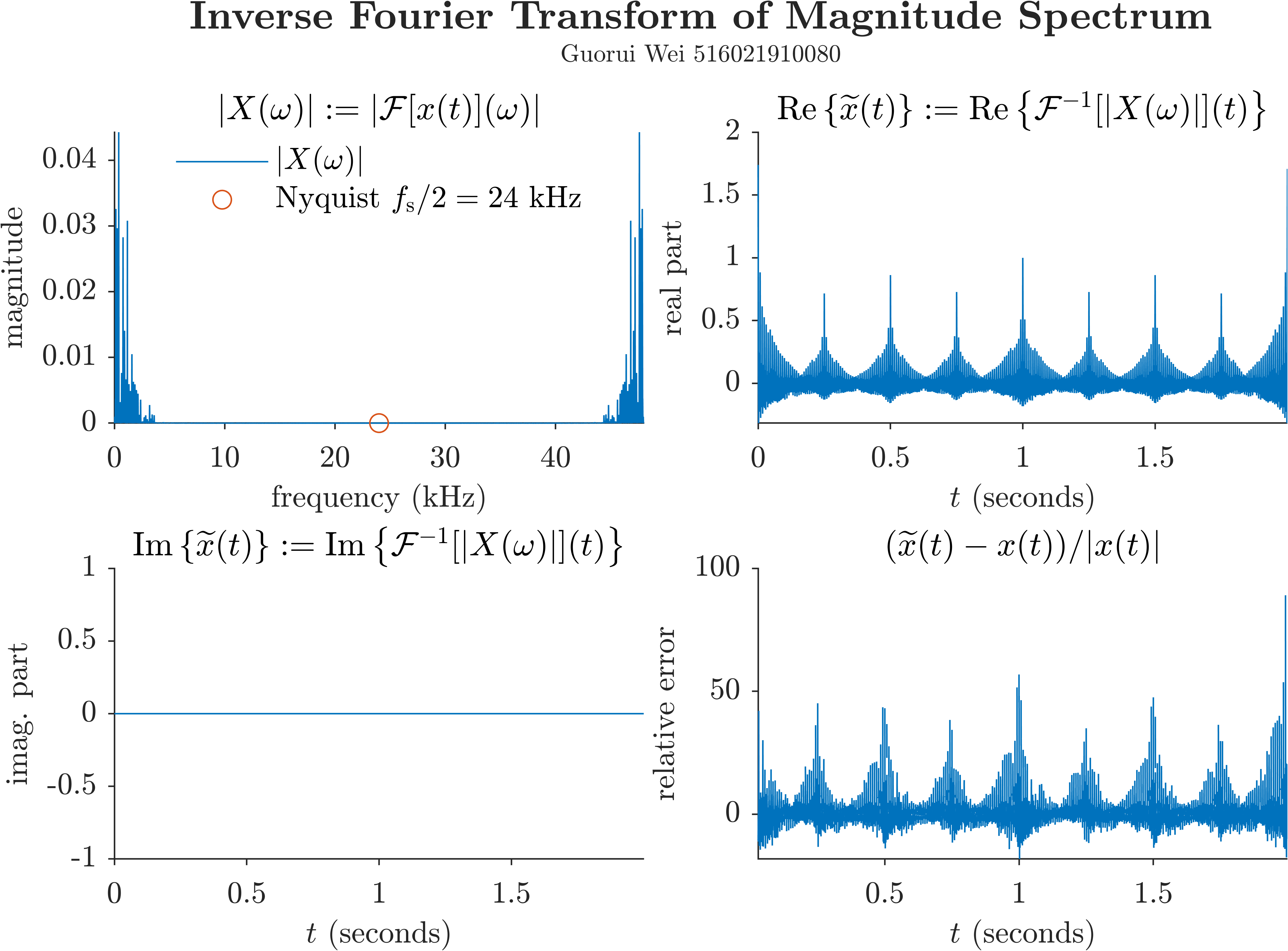


图3.4 对原始信号用算法3.1求出的数值CTFT的幅度谱，再用算法3.2求数值ICTFT的结果. 左上图是对原始信号用算法3.1求出的数值CTFT的幅度谱. 对这CTFT幅度谱用算法3.2求出的数值ICTFT的实部和虚部分别示于右上图和左下图，可见这数值ICTFT是实的（与命题3.4一致）. 这数值ICTFT对原始信号的相对误差示于右下图，在实际计算中跳过了那些使得原始信号的绝对值很小的时刻；可见与原始信号差别非常大.

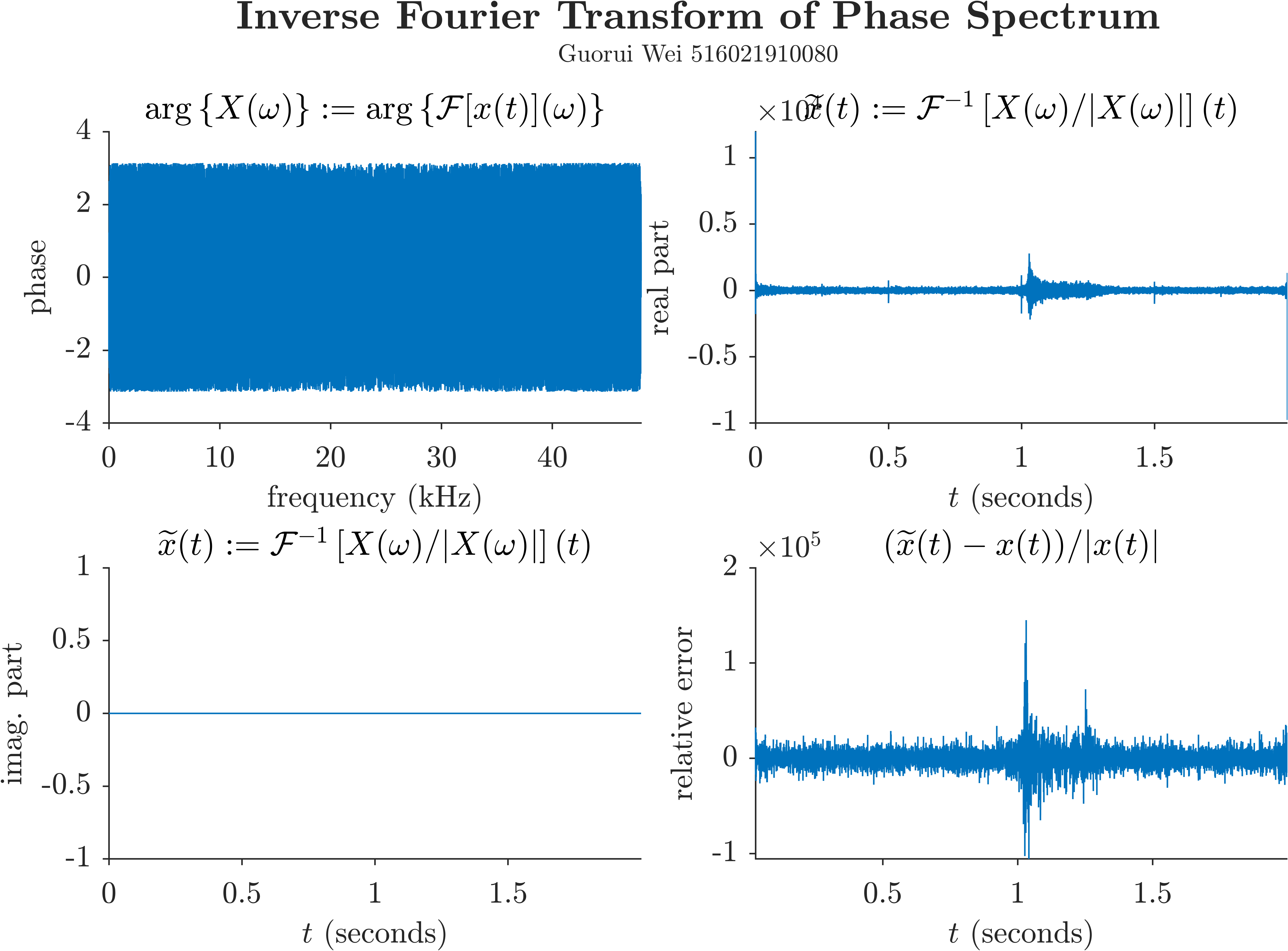


图3.5 对原始信号用算法3.1求出的数值CTFT的相位谱，再用算法3.2求数值ICTFT的结果. 左上图是对原始信号用算法3.1求出的数值CTFT的相位谱. 对这CTFT相位谱用算法3.2求出的数值ICTFT的实部和虚部分别示于右上图和左下图，可见这数值ICTFT是实的（与命题3.4一致）. 这数值ICTFT对原始信号的相对误差示于右下图，在实际计算中跳过了那些使得原始信号的绝对值很小的时刻；可见差别非常大.

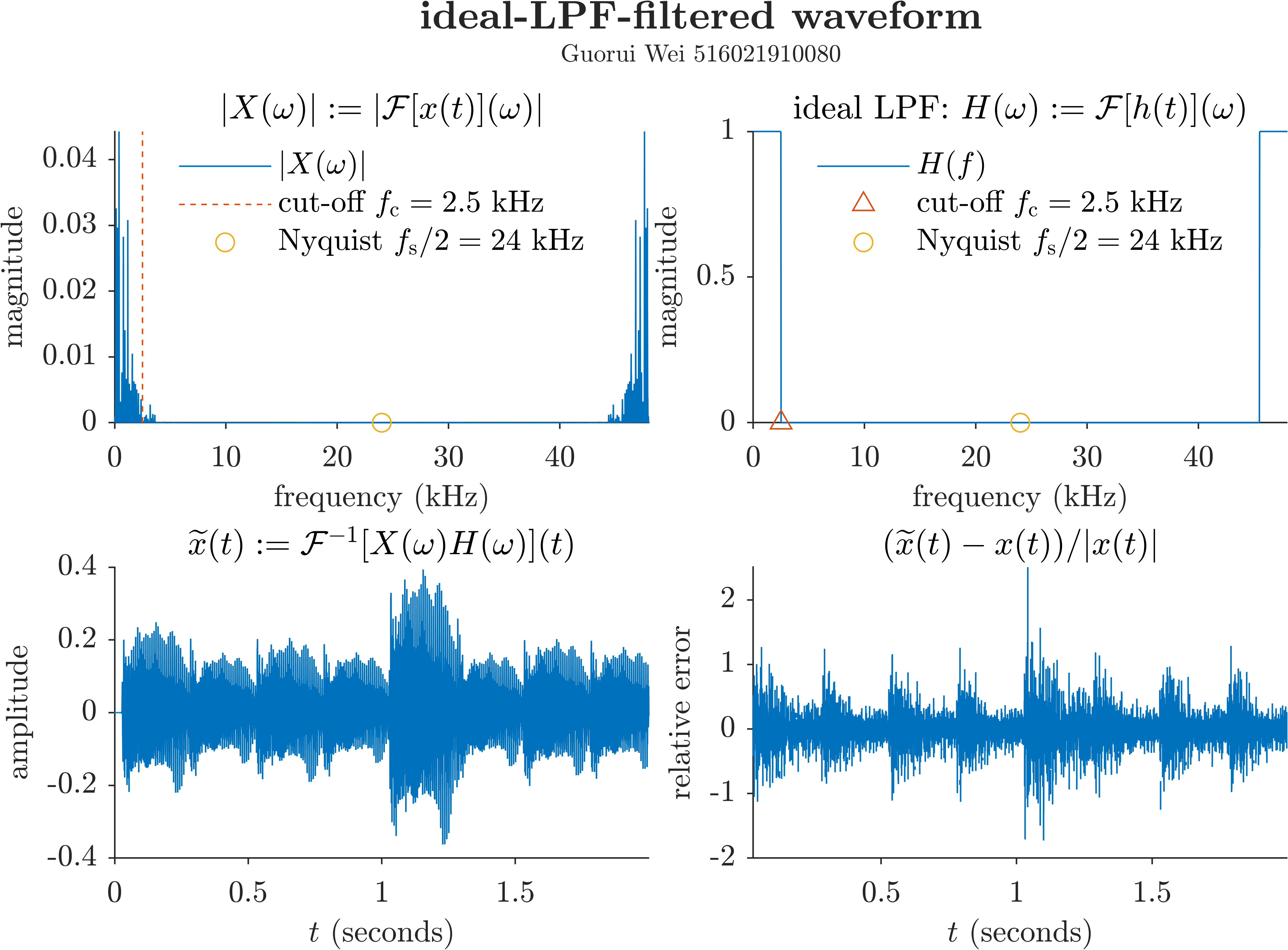


图3.6 原始信号经过一个理想低通滤波器（low-pass filter, LPF）后的波形和与原始信号的差别. 左上图展示了对原始信号用算法3.1求出的数值CTFT的幅度谱，理想LPF的截止频率（2.5 kHz），以及原始音频信号的采样率对应的Nyquist频率（kHz）. 右上图展示了理想LPF的频率相应，它是实的. 左下图展示了原始信号经过一个理想低通滤波器（low-pass filter, LPF）后的波形，可见与原始信号形状很相似. 右下图展示了滤波后的信号对原始信号的相对误差，在实际计算中跳过了那些使得原始信号的绝对值很小的时刻；可见，虽然滤波后的信号的形状与原始信号很相似，但在许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

## 项目小结

本项目利用点DFT与CTFT的关系（），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（），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（），时移性质（）和奇偶虚实性（），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.

# 项目2：语音采样器（Speech Sampler）

## 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 Hz)

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.

## 数据和方法

### 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）

其中

这种信号可通过计算机软件（例如Matlab）方便地生成.



图4.1 本项目使用的连续时间信号该信号仅含3个等幅且初相不同的单音频率分量，即 因此具有400 μs周期. 本图展示了在两个周期上的图像.

### 冲激串采样与理想带限内插

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

相乘，得采样后信号其频谱

可见，就是的等幅周期拓展.

若是带限(band-limited)的，即的频谱满足

则只需将取样后的信号通过一个截止频率为幅度为的理想低通滤波器（频率响应记为），就可恢复出原始信号 理想低通滤波器的冲激响应

上式中的sinc函数定义为

得到的重建信号

若取 则有

特别地，

可见，重建信号经过所有采样点，可将视作在插值节点下，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) 设带限信号满足（），其等间隔样值若采样频率满足，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（），可精确重建.

注4.2 (时域有限信号的重建) 重建公式（）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（）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（）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（），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（）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（）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（）可用于内插（interpolation），而不用于外推（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（），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（）在采样时间区间上精确重建原始信号.

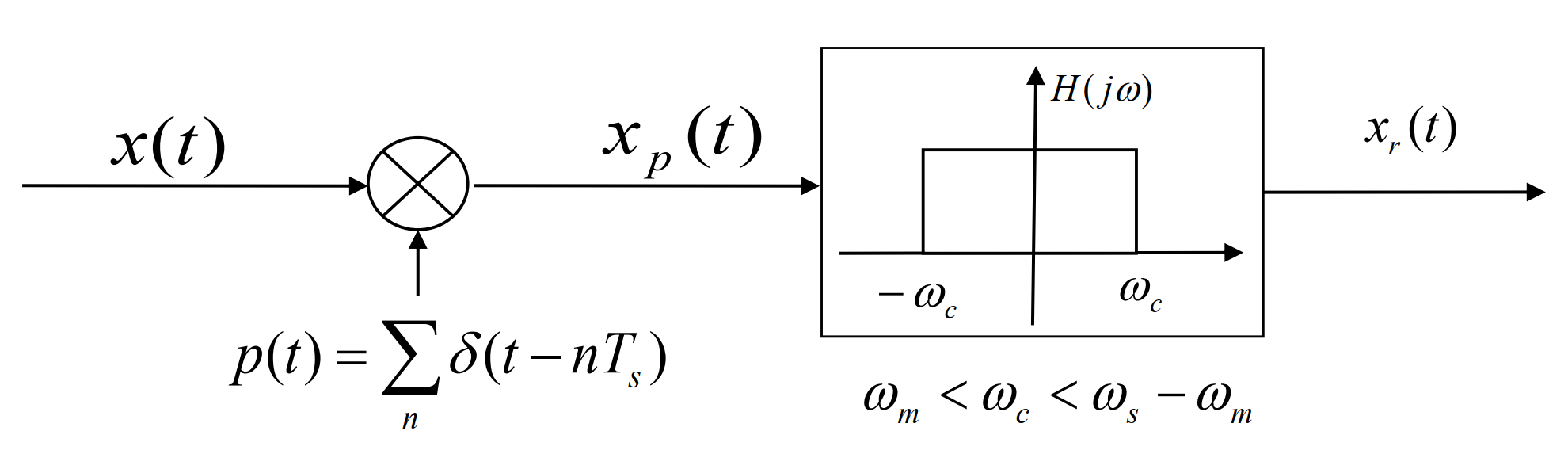


图4.2 理想采样与重建原理图. 是带限信号(band-limited signal)，当时，的频谱 是取样函数(sampling function)，在冲激串采样中取为周期冲激串，取样周期为 即取样频率 取样后的信号通过一个截止频率为幅度为的理想低通滤波器，得到重建信号 若则有 图源：上海交通大学2022年春季学期ICE2301课程教学用课件“signal&system(v3.1)-ch4-FT-频域分析.pdf”第66页（刘伟，2022）.

### 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其等时间间隔样值

其中 下面建立其理想带限内插.

记则其等时间间隔样值

于是的理想带限内插（）成为

若满足定理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），则也满足定理4.1的条件（因为具有相同的幅频特性），从而

为此，规定

为时域有限的连续时间信号的理想带限内插.

定义连续时间信号的样值序列和取样函数序列：

即

则（）可写成

现在用等时间间隔样值理想带限内插到上，试图重建原始信号定义重建信号序列

则由（）得

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，则和原始信号在相应时刻的值相等.

### 信号重建质量的评价

由（）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（）对非取样时刻原始信号的恢复情况. 注4.3指出，重建公式（）不宜用于外推，故不必考察（）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

其中是重建时刻，是利用理想带限内插重建公式（）对原始信号在采样时间区间上的重建信号. 定义（）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（）的分母很小的点.

## 结果和讨论

### 重建的信号

对仅含10，5，2.75 kHz三个频率分量的原始信号（），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（）获得重建信号在重建时刻上的值. 用（）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（）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（）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（）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（）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kHz）的单音分量，5 kHz的单音分量被混叠成0.49 kHz（理论值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（）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kHz）的单音分量，5 kHz的单音分量被混叠成0.47 kHz（理论值kHz）的单音分量，2.5 kHz的单音分量被混叠成0.23 kHz（理论值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.

### 讨论

对图4.3所示的实验结果，可得出以下观察.

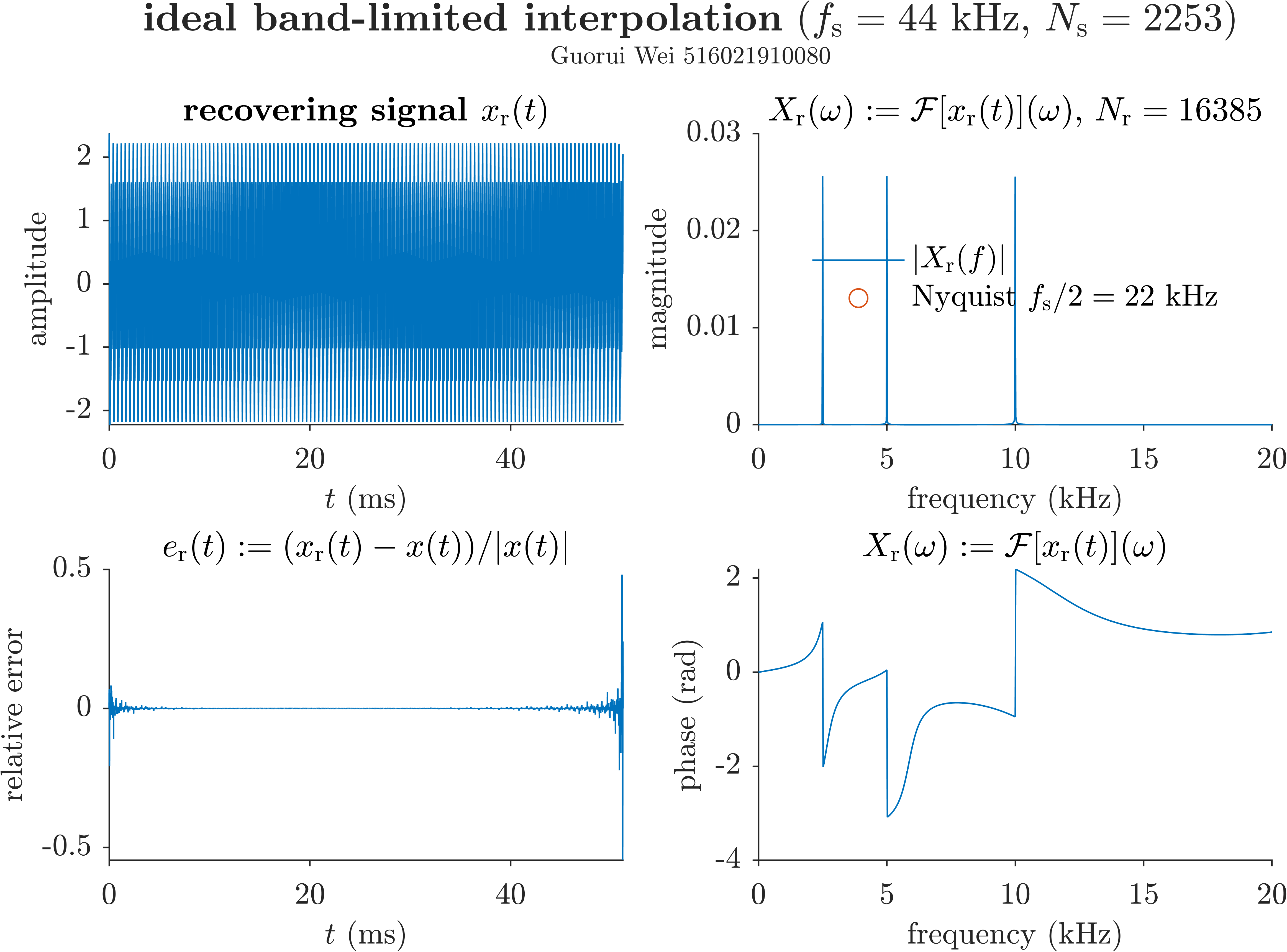
1. 即使采样频率高于原始信号对应的Nyquist频率，也不能通过理想带限内插公式（）在采样时间区间内精确重建原始信号. 这是因为，使用有限的采样时间区间，意味着重建的实际上是作为原始信号一部分的一个时域有限信号，而时域有限信号通常是频域无限的，不满足定理4.1的条件. 这与注4.3的描述一致.

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.

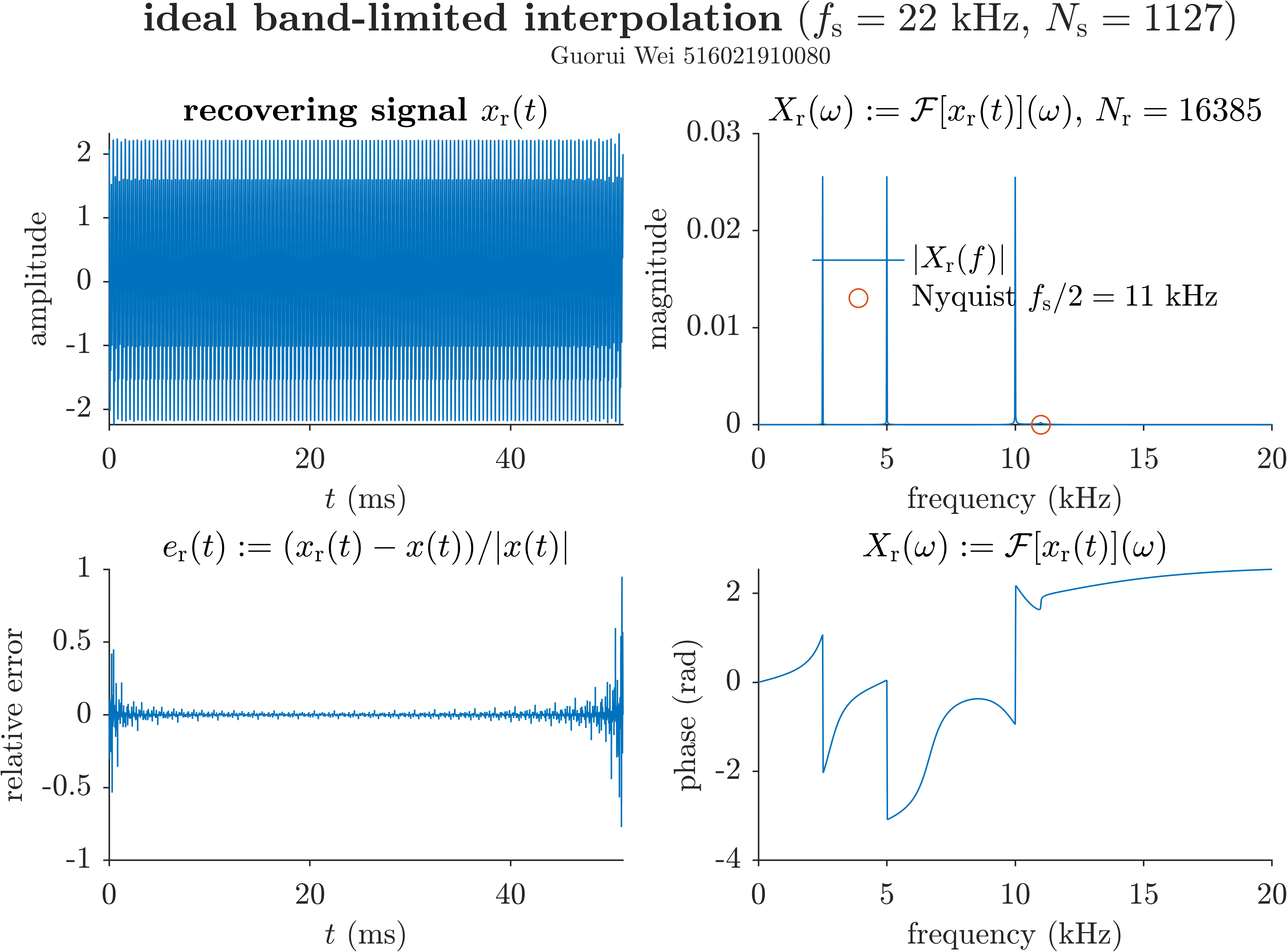
3. 当采样频率高于原始信号对应的Nyquist频率时，重建误差在采样时间区间的中心附近较小，而在接近区间边缘处升高，类似Gibbs现象.

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.

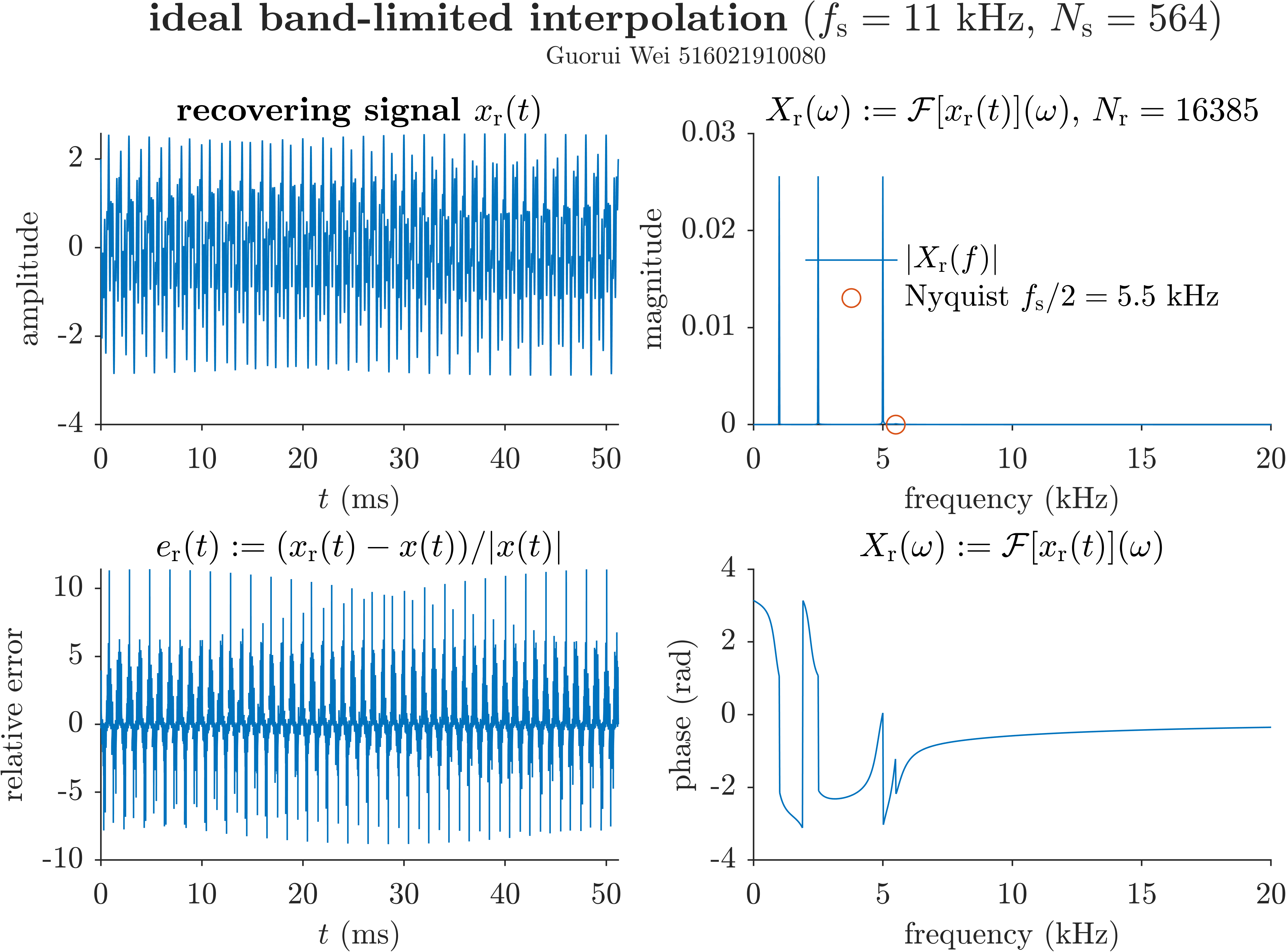
5. 利用时域无限信号在有限采样时间区间上的等间隔样值，通过理想带限内插公式（）获得的重建信号，与原始信号相比，其频率在采样频率的二分之一附近的频率分量可能会略有增强. 原因可能是，通过有限采样时间区间上的样值按（）恢复原始信号，隐含采样时间区间外的样值全为零，相当于在采样时间区间边缘引入了一个从非零到零的跳变，这对应一个频率约等于采样频率的二分之一的分量.



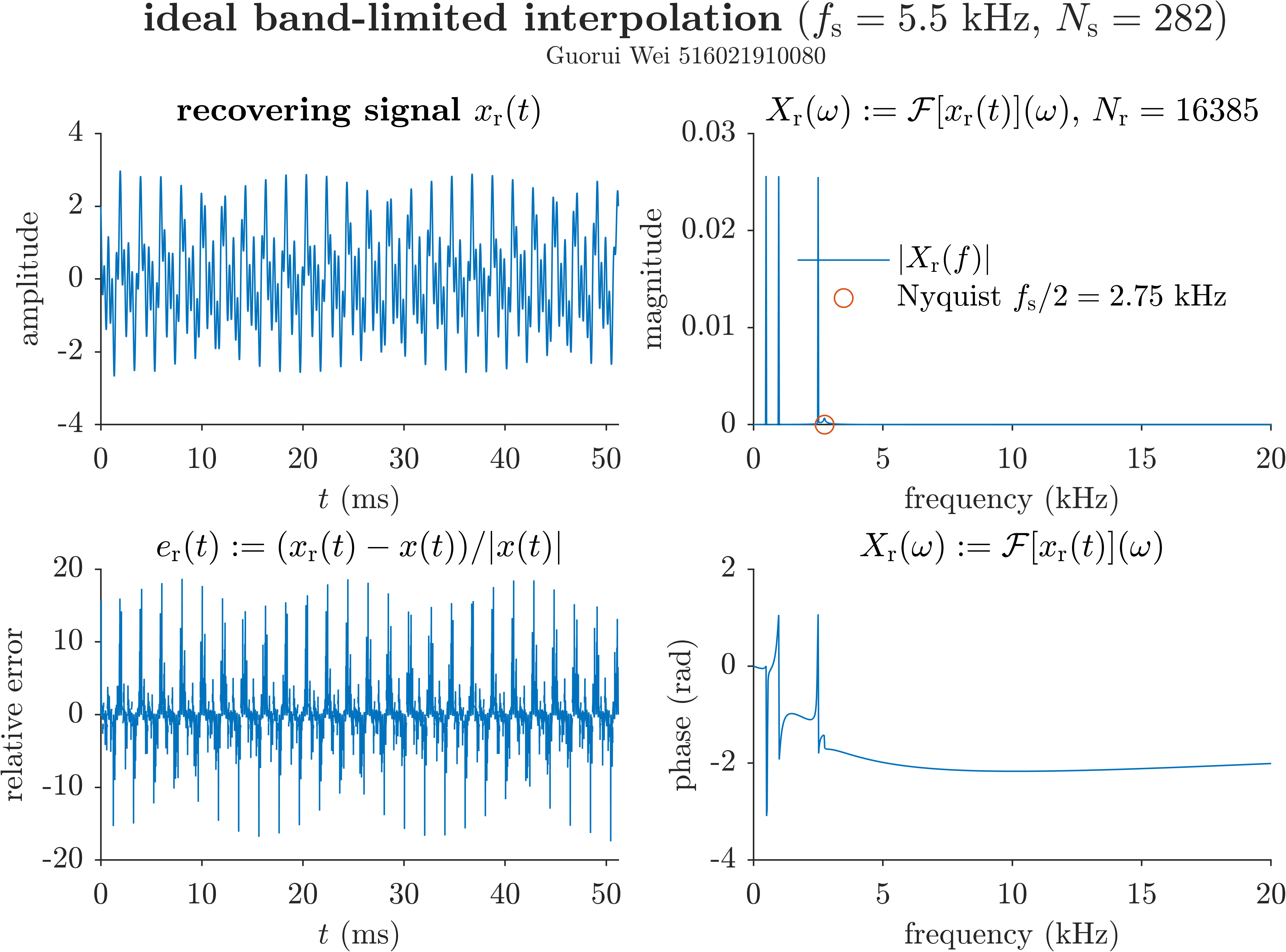
（a）



（b）



（c）



（d）



（e）

图4.3 重建的信号. 对原始信号（）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（）计算重建信号在重建时刻上的值. 用（）计算重建相对误差.

## 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（）. 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（）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（）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（）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（）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

1. 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/>.

* 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 orginal\_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\_orginal\_data\_and\_its\_time-domain\_transformation.emf",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 51 | exportgraphics(t\_TCL\_ampl,"..\\doc\\fig\\proj1\\Fig\_2a\_CTFT\_ampl\_of\_orginal\_data\_and\_its\_time-domain\_transformation.png",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 52 | % exportgraphics(t\_TCL\_pha,"..\\doc\\fig\\proj1\\Fig\_2b\_CTFT\_phase\_of\_orginal\_data\_and\_its\_time-domain\_transformation.emf",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 53 | exportgraphics(t\_TCL\_pha,"..\\doc\\fig\\proj1\\Fig\_2b\_CTFT\_phase\_of\_orginal\_data\_and\_its\_time-domain\_transformation.png",'Resolution',800,'ContentType','auto','BackgroundColor','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','BackgroundColor','none','Colorspace','rgb'); |
| 105 | exportgraphics(t\_TCL,"..\\doc\\fig\\proj1\\Fig\_1\_orginal\_wave\_and\_its\_time-domain\_transformation.png",'Resolution',800,'ContentType','auto','BackgroundColor','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 516021910080",'Interpreter','latex'); |
| 128 | set(t\_title\_s,'FontSize',8); |
| 129 | % |
| 130 | % exportgraphics(t\_TCL,"..\\doc\\fig\\proj1\\Fig\_3a\_ICTFT\_original.emf",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 131 | exportgraphics(t\_TCL,"..\\doc\\fig\\proj1\\Fig\_3a\_ICTFT\_original.png",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 132 |  |
| 133 | %% Fig.3b inverse CTFT of magnitude spectrum |
| 134 |  |
| 135 | figure('Name',"Fig.3b ICTFT: magnitude spectrum"); |
| 136 | t\_TCL = tiledlayout(2,2,"TileSpacing","tight","Padding","tight"); |
| 137 | % |
| 138 | t\_axes = nexttile(t\_TCL,1); |
| 139 | plot(t\_axes,omega\_ori/2/pi/1000,mag\_spectrum,'-', ... |
| 140 | "DisplayName",'$|X(\omega)|$'); |
| 141 | hold on |
| 142 | plot(t\_axes,F\_s/2/1000,0,'o', ... |
| 143 | "DisplayName",sprintf("Nyquist $f\_{\\rm s}/2 = %g$ kHz",F\_s/2/1000)); |
| 144 | hold off |
| 145 | set(t\_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out','XLimitMethod','tight'); |
| 146 | legend(t\_axes,"Location",'best','Interpreter','latex',"Box","off",'FontSize',10); |
| 147 | xlabel(t\_axes,"frequency (kHz)",FontSize=10,Interpreter="latex"); |
| 148 | ylabel(t\_axes,"magnitude","FontSize",10,"Interpreter","latex"); |
| 149 | title(t\_axes,"$|X(\omega)| := |\mathcal{F}[x(t)](\omega)|$",'Interpreter','latex'); |
| 150 | % |
| 151 | t\_axes = nexttile(t\_TCL,2); |
| 152 | plot(t\_axes,t\_mag\_IFT,real(x\_mag\_IFT),'-',"DisplayName",'ICTFT (real) of magnitude spectrum'); |
| 153 | set(t\_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out','XLimitMethod','tight'); |
| 154 | xlabel(t\_axes,"$t$ (seconds)",FontSize=10,Interpreter="latex"); |
| 155 | ylabel(t\_axes,"real part","FontSize",10,"Interpreter","latex"); |
| 156 | title(t\_axes,"${\rm Re} \left\{ \widetilde{x}(t) \right\} := {\rm Re} \left\{ \mathcal{F}^{-1}[|X(\omega)|](t) \right\}$",'Interpreter','latex'); |
| 157 | % |
| 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",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 178 | exportgraphics(t\_TCL,"..\\doc\\fig\\proj1\\Fig\_3b\_ICTFT\_mag\_spectrum.png",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 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",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 219 | exportgraphics(t\_TCL,"..\\doc\\fig\\proj1\\Fig\_3c\_ICTFT\_pha\_spectrum.png",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 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 | "DisplayName",sprintf("cut-off $f\_{\\rm c} = %g$ kHz",omega\_c/2/pi/1000)); |
| 232 | plot(t\_axes,F\_s/2/1000,0,'o', ... |
| 233 | "DisplayName",sprintf("Nyquist $f\_{\\rm s}/2 = %g$ kHz",F\_s/2/1000)); |
| 234 | hold off |
| 235 | set(t\_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out','XLimitMethod','tight'); |
| 236 | legend(t\_axes,"Location",'best','Interpreter','latex',"Box","off",'FontSize',10); |
| 237 | xlabel(t\_axes,"frequency (kHz)",FontSize=10,Interpreter="latex"); |
| 238 | ylabel(t\_axes,"magnitude","FontSize",10,"Interpreter","latex"); |
| 239 | title(t\_axes,"$|X(\omega)| := |\mathcal{F}[x(t)](\omega)|$",'Interpreter','latex'); |
| 240 | % |
| 241 | t\_axes = nexttile(t\_TCL,2); |
| 242 | plot(t\_axes,omega\_ori/2/pi/1000,LPF\_spec,'-', ... |
| 243 | "DisplayName",'$H(f)$'); |
| 244 | hold on |
| 245 | plot(t\_axes,omega\_c/2/pi/1000,0,'^', ... |
| 246 | "DisplayName",sprintf("cut-off $f\_{\\rm c} = %g$ kHz",omega\_c/2/pi/1000)); |
| 247 | plot(t\_axes,F\_s/2/1000,0,'o', ... |
| 248 | "DisplayName",sprintf("Nyquist $f\_{\\rm s}/2 = %g$ kHz",F\_s/2/1000)); |
| 249 | hold off |
| 250 | set(t\_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out','XLimitMethod','tight'); |
| 251 | legend(t\_axes,"Location",'best','Interpreter','latex',"Box","off",'FontSize',10); |
| 252 | xlabel(t\_axes,"frequency (kHz)",FontSize=10,Interpreter="latex"); |
| 253 | ylabel(t\_axes,"magnitude","FontSize",10,"Interpreter","latex"); |
| 254 | title(t\_axes,sprintf("ideal LPF: $H(\\omega) := \\mathcal{F}[h(t)](\\omega)$"),'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",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 276 | exportgraphics(t\_TCL,"..\\doc\\fig\\proj1\\Fig\_4\_LPF\_filtered\_signal.png",'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 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-strlength(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",10,'Box','off','TickDir','out','XLimitMethod','tight'); |
| 344 | title(axes\_amp,str\_title,'Interpreter','latex'); |
| 345 | % |
| 346 | plot(axes\_pha,omega(1:N\_half)/2/pi/1000,angle(X(1:N\_half)),'-', ... |
| 347 | "DisplayName",str\_title); |
| 348 | set(axes\_pha,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out','XLimitMethod','tight'); |
| 349 | title(axes\_pha,str\_title,'Interpreter','latex'); |
| 350 | end |
| 351 |  |
| 352 | %% ICTFT by IDFT |
| 353 |  |
| 354 | function [t,x] = ICTFT(omega,X,t0) |
| 355 | % inverse continuous-time Fourier transform, approximated by IDFT. |
| 356 | % CAVEAT: MUST be sampled at equal intervals! |
| 357 | arguments |
| 358 | omega |
| 359 | X |
| 360 | t0 = 0; |
| 361 | end |
| 362 |  |
| 363 | if(~iscolumn(omega)) |
| 364 | omega = omega.'; |
| 365 | end |
| 366 | if(~iscolumn(X)) |
| 367 | X = X.'; |
| 368 | end |
| 369 |  |
| 370 | [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 |  |

* 1. 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 origina 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","$|X\_{\rm o}(f)|$"); |
| 33 | set(t\_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out'); |
| 34 | xlim(t\_axes,[0,2\*max(F\_ori)/1000]); |
| 35 | xlabel(t\_axes,"frequency (kHz)",FontSize=10,Interpreter="latex"); |
| 36 | ylabel(t\_axes,"magnitude","FontSize",10,"Interpreter","latex"); |
| 37 | title(t\_axes,"$X\_{\rm o}(\omega) := \mathcal{F}[x\_{\rm o}(t)](\omega)$",'Interpreter','latex'); |
| 38 | % CTFT of original signal (pha.) |
| 39 | t\_axes = nexttile(t\_TCL,2); |
| 40 | plot(t\_axes,omega\_ori/2/pi/1000,angle(CTFT\_ori),'-', ... |
| 41 | "DisplayName","${\rm arg}\{X\_{\rm o}(f)\}$"); |
| 42 | set(t\_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out','XLim',[0,2\*max(F\_ori)/1000]); |
| 43 | xlabel(t\_axes,"frequency (kHz)",FontSize=10,Interpreter="latex"); |
| 44 | ylabel(t\_axes,"phase (rad)","FontSize",10,"Interpreter","latex"); |
| 45 | title(t\_axes,"$X\_{\rm o}(\omega) := \mathcal{F}[x\_{\rm o}(t)](\omega)$",'Interpreter','latex'); |
| 46 | % original signal |
| 47 | t\_axes = nexttile(t\_TCL,3,[1 2]); |
| 48 | plot(t\_axes,t\_o(1:2\*F\_s\*cycle\_ori+1)\*1e6,x\_o(1:2\*F\_s\*cycle\_ori+1),'-', ... |
| 49 | "DisplayName","$x\_{\rm o}(t)$"); |
| 50 | set(t\_axes,"YDir",'normal',"TickLabelInterpreter",'latex',"FontSize",10,'Box','off','TickDir','out','XLimitMethod','tight'); |
| 51 | xlabel(t\_axes,"$t$ $(\rm \mu s)$",FontSize=10,Interpreter="latex"); |
| 52 | ylabel(t\_axes,"amplitude","FontSize",10,"Interpreter","latex"); |
| 53 | title(t\_axes,sprintf("{\\bf original signal $x\_{\\rm o}(t)$} (periodic, $T = %g$ ${\\rm \\mu s}$)",1e6/min(F\_ori)),'Interpreter','latex'); |
| 54 | % |
| 55 | [~,t\_title\_s] = title(t\_TCL,sprintf("\\bf original signal (periodic)"),"Guorui Wei 516021910080",'Interpreter','latex'); |
| 56 | set(t\_title\_s,'FontSize',8); |
| 57 | % |
| 58 | exportgraphics(t\_TCL,sprintf("..\\doc\\fig\\proj2\_2\\Fig\_0\_original\_signal.emf"),'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 59 | exportgraphics(t\_TCL,sprintf("..\\doc\\fig\\proj2\_2\\Fig\_0\_original\_signal.png"),'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 60 |  |
| 61 | %% Recover signal by ideal band-limited interpolation. |
| 62 |  |
| 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 [MUST be geometric series!] |
| 176 | phi\_ori = [0,0];        % initial phase of components of origina signal |
| 177 | end |
| 178 |  |
| 179 | cycle\_ori = 1/min(F\_ori);                       % cycle of original signal |
| 180 | t\_n = linspace(t\_s0,t\_s0+tau\_s,F\_s\*tau\_s+1).';  % sampling time |
| 181 | x\_n = x\_ori(t\_n,F\_ori,phi\_ori);                 % sampling values |
| 182 | t\_r = linspace(t\_r0,t\_r0+tau\_r,F\_r\*tau\_r+1).';  % recovering time |
| 183 | x\_r = ibli(t\_n,x\_n,t\_r);                        % recovering values |
| 184 | x\_r\_ori = x\_ori(t\_r,F\_ori,phi\_ori);             % original values at 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",'FontSize',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("$X\_{\\rm r}(\\omega) := \\mathcal{F}[x\_{\\rm r}(t)](\\omega),$ $N\_{\\rm r} = %g$",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","${\rm arg}\{X\_{\rm r}(f)\}$"); |
| 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",'FontSize',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,"$X\_{\rm r}(\omega) := \mathcal{F}[x\_{\rm r}(t)](\omega)$",'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\_limited\_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\_band\_limited\_interp.png",num2str(F\_s/1000)),'Resolution',800,'ContentType','auto','BackgroundColor','none','Colorspace','rgb'); |
| 239 | end |
| 240 |  |

1. https://musiclab.chromeexperiments.com/Song-Maker/ [↑](#footnote-ref-1)
2. continuous functions that are only nonzero on a closed subinterval of [↑](#footnote-ref-2)
3. https://ww2.mathworks.cn/help/matlab/ref/fft.html [↑](#footnote-ref-3)
4. https://ww2.mathworks.cn/help/matlab/ref/ifft.html [↑](#footnote-ref-4)