

基于矩形窗的短时傅里叶变换实现与音频时频分析

3240102873 胡东东

摘要：傅里叶变换能够将时域信号转换至频域，但无法反映频率分量随时间的变化规律，不适合非平稳信号的分析。短时傅里叶变换（STFT）通过在时间轴上滑动窗函数并对加窗信号进行傅里叶变换，实现了信号的时频联合表示。本文基于矩形窗实现了短时傅里叶变换算法，利用 MATLAB 生成了线性扫频信号与多音脉冲信号，绘制了它们的幅度谱并对比了二者的频谱特性。针对不同窗长，绘制了两个信号的时频图，探讨了窗长对时间分辨率与频率分辨率的影响。在此基础上，选取一段 MP3 音乐，利用自编的矩形窗 STFT 提取实时频谱，完成了动态频谱柱状图的动画显示，直观展现了音乐信号的时频变化过程。实验结果表明，矩形窗 STFT 能够有效揭示频率随时间的变化趋势，窗长的选择决定了时频分辨率的取舍，该方法为音频信号的可视化分析提供了一种简单而有效的工具。

关键词： 短时傅里叶变换；矩形窗；时频分析；频谱可视化；音频处理

一、引言

傅里叶变换是信号处理领域的重要工具，它将信号从时域映射到频域，使我们能够通过频谱特性理解信号的频率成分。然而，传统傅里叶变换是对整个时间轴上的信号进行全局积分，其结果仅反映信号在整个观测时段内的平均频率特征，无法揭示各频率分量随时间的变化情况。许多实际信号如语音、音乐、雷达回波等都呈现明显的非平稳特性，其频谱随时间是动态变化的。因此，单靠傅里叶变换很难分析信号中频率分量的起止时间、持续时间及瞬态事件，这限制了它在时变信号分析中的应用。

为解决这一问题，短时傅里叶变换（Short-Time Fourier Transform, STFT）被提出并得到广泛应用。STFT 的基本思想是在信号上滑动一个有限长度的窗函数，每一时刻截取窗口内的信号段进行傅里叶变换，从而获得该时刻附近的局部频谱。随着窗口在时间轴上移动，就可得到信号在不同时刻的频谱分布，即信号的时频表示。窗函数的选择和窗长的大小直接决定了时间分辨率和频率分辨率之间的平衡：短窗能精确捕捉信号的瞬态变化但频率分辨能力下降，长窗则能更好地区分相邻频率但时间定位变得模糊。

本文基于矩形窗实现了短时傅里叶变换算法，并将其应用于合成信号和真实音频信号的时频分析。首先，生成了两个具有不同时频特性的测试信号——线性扫频信号和多

音叠加瞬态脉冲信号，通过绘制它们的傅里叶幅度谱，说明传统傅里叶变换的局限性。其次，编写了矩形窗 STFT 函数，并利用该函数绘制了两种信号在不同窗长下的时频图，分析窗口长度对时频分辨率的影响规律。最后，选取一段 MP3 音乐，利用自己编写的 STFT 提取实时频谱，制作了动态频谱柱状图动画，直观展示了音乐播放过程中频率能量的时间演化。所有程序均在 MATLAB 环境下实现，信号采样率统一为 4410 Hz。实验结果表明，基于矩形窗的 STFT 能有效实现非平稳信号的时频可视化，为后续进一步研究窗函数优化和音频特征分析提供了基础。

二、短时傅里叶变换的基本原理

短时傅里叶变换 (Short-Time Fourier Transform, STFT) 是分析非平稳信号的一种基本时频方法。其核心思想是：在信号上滑动一个有限长度的窗函数，每次截取出以当前时刻为中心的一小段信号，对这一小段信号进行傅里叶变换，得到该时刻附近的局部频谱。随着窗函数在时间轴上不断移动，就可以获得信号在不同时刻的频谱分布。

对于一个连续时间信号 $x(t)$ ，其短时傅里叶变换定义为

$$STFT_x(t, f) = \int_{-\infty}^{\infty} x(\tau) w(\tau - t) e^{-j2\pi f\tau} d\tau$$

其中 $w(t)$ 为窗函数，它在时域上以 t 为中心截取信号，使得频谱只反映窗内信号段的频率特性。当窗函数在时间轴上滑动时， $STFT_x(t, f)$ 给出了信号在每一个时刻 t 附近的频率成分，实现了时间和频率的联合表示。

在实际应用中，信号和系统都是离散的。将连续信号 $x(t)$ 以采样周期 T_s 转换成数字序列 $x[n]$ ，窗函数也相应离散化为 $w[n]$ ，则离散短时傅里叶变换可表示为

$$STFT_x[m, k] = \sum_{n=-\infty}^{\infty} x[n] w[n - m] e^{-j\frac{2\pi}{N}kn}$$

式中 m 为时间索引，对应窗口的位置； k 为频率索引； N 为窗口长度。通常并不计算窗口在整个信号长度上的每一个采样点，而是以一定的步长（帧移）向前滑动，以减小计算量。对每一帧加窗后的数据做 N 点快速傅里叶变换 (FFT)，即可得到该帧的复频谱。将所有帧的频谱沿时间轴排列，便构成了信号的时频谱矩阵。

本文选用的窗函数为矩形窗 (Rectangular Window)，其离散形式为

$$w[n] = \begin{cases} 1, & 0 \leq n < L \\ 0, & \text{其他} \end{cases}$$

其中 L 为窗口长度。矩形窗具有最窄的主瓣宽度，能提供最高的频率分辨率，但其

旁瓣较高，容易造成频谱泄漏。实际分析中，窗口长度 L 的选择直接影响时频分辨率。根据傅里叶变换的不确定性原理，窗口越短，时间定位越精确，但频率分辨率越低；反之，窗口越长，频率分辨率越高，但时间定位变得模糊。因此，在实际应用中需要根据信号特征在时间分辨率和频率分辨率之间做出折衷。

将每一帧的频谱幅值绘制成二维图像，横轴为时间，纵轴为频率，颜色深浅表示幅度大小，即可得到信号的时频图。时频图直观地展示了信号中各频率分量的起始、持续时间和强度变化，是分析非平稳信号的有力工具。本文后续实验将基于上述原理，利用自编的矩形窗 STFT 函数，分别对合成信号和真实音频信号进行时频分析与可视化。

三、算法设计与实现

(1) 信号生成与预处理

为了验证传统傅里叶变换在分析非平稳信号时的局限性，并为后续短时傅里叶变换提供测试对象，首先构造了两段具有不同时频特性的离散时间信号。信号的采样率统一设定为 4410 Hz，信号时长均为 2 秒，即每段信号包含 $N = 8820$ 个采样点。

信号 1 为一线性扫频信号（Chirp 信号），其瞬时频率随时间线性增加。设起始频率 $f_0 = 200\text{Hz}$ ，终止频率 $f_1 = 2000\text{Hz}$ ，扫频周期等于信号时长。该信号模拟了频率连续快速变化的场景，其表达式为

$$x_1[n] = \sin\left(2\pi\left(f_0 nT_s + \frac{f_1 - f_0}{2T}(nT_s)^2\right)\right), n = 0, 1, \dots, N-1$$

其中 $T_s = 1/4410\text{s}$ 为采样间隔， $T = 2\text{s}$ 。在 MATLAB 中通过 chirp 函数直接生成。

信号 2 为多频混合叠加瞬态高频脉冲的信号。其背景由两个稳定的正弦分量组成，频率分别为 400 Hz 和 1500 Hz，幅度分别为 0.5 和 0.3。在 $0.8 \leq t < 1.0$ 秒的区间内，额外叠加一个 2500 Hz、幅度为 0.6 的正弦脉冲，以模拟局部突发的频率分量。该信号的数学表示为

$$x_2[n] = 0.5\sin(2\pi \cdot 400 \cdot nT_s) + 0.3\sin(2\pi \cdot 1500 \cdot nT_s) + s[n]$$
$$s[n] = \begin{cases} 0.6\sin(2\pi \cdot 2500 \cdot nT_s), & 0.8 \leq nT_s < 1.0 \\ 0, & \text{其他} \end{cases}$$

该信号同时包含了稳态频率和瞬态频率成分，可有效检验时频分析工具对稳定谱线和短时脉冲的分辨能力。

两段信号生成后，分别计算其离散傅里叶变换（DFT），得到单边幅度谱，频率轴范

围为0 ~ 2205Hz。幅度谱通过下式归一化：

$$|X[k]| = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \right|, k = 0, 1, \dots, \left[\frac{N}{2}\right]$$

并除去直流分量外的其他分量乘以 2 以得到正确的单边幅度。

生成的两组信号及其幅度谱被保存为 signal1.mat 和 signal2.mat 文件，供后续 STFT 分析程序直接加载使用。该预处理流程不涉及滤波或重采样，仅完成从数学定义到数字序列的转化以及频域初步分析，为后面几节的实验奠定了基础。

(2) 矩形窗 STFT 函数的编写

在完成测试信号的构造与频谱初步分析后，第二部分的核心任务是实现短时傅里叶变换算法。为此，基于短时傅里叶变换的基本原理，我编写了函数 my_stft，专门用于计算信号在矩形窗下的时频谱。

函数接口定义为：`[S, f, t] = my_stft(x, fs, win_len, hop)`

其中各输入参数的含义分别为： x 为待分析的离散时间信号， fs 为信号的采样率(Hz)， win_len 为矩形窗的长度（以采样点数计）， hop 为相邻帧之间的位移量（帧移）。函数的输出包含三个部分： S 为一个复数矩阵，其行对应频率，列对应不同的时间帧，构成信号的时频谱； f 为频率向量，表示矩阵每一行所对应的频率值； t 为时间向量，给出每一帧的起始时间。

实现过程完全遵循短时傅里叶变换的离散定义。首先，将输入信号转换为列向量以统一处理。根据信号总长度 N 和窗长、帧移，确定所有可能的帧起始位置：

`frame_starts = [1, 1 + hop, 1 + 2hop, ... ]`, 满足帧尾部不超过信号长度。

每一帧对应信号中从起始点开始、长度为 win_len 的一段数据。对该段数据乘以矩形窗，矩形窗通过 MATLAB 的 `rectwin` 函数生成，窗内所有采样点权重均为 1。加窗后执行 win_len 点的快速傅里叶变换（FFT），得到该帧的复频谱。

考虑到实信号的傅里叶变换具有共轭对称性，为了减少存储和计算量，仅保留频谱的前 $win_len/2+1$ 个点，对应频率从 0 到奈奎斯特频率 $f_s/2$ 的单边谱。对单边谱内部的频率分量（不含直流和奈奎斯特频率），将幅值乘以 2，以便恢复信号的真实幅度。最后，将结果除以窗长进行归一化，以保持不同窗长下幅值的可比性。归一化后的复频谱存入矩阵 S 的对应列中。

完成所有帧的计算后，根据窗长和采样率生成频率向量：

$$f[k] = \frac{k \cdot f_s}{win_len}, k = 0, 1, \dots, \frac{win_len}{2},$$

并根据帧起始索引给出每帧对应的时间轴：

$$t[m] = \frac{frame_starts[m] - 1}{f_s}, m = 1, 2, \dots, num_frames.$$

由此得到的 S 矩阵、f 和 t 向量，可直接用于绘制时频图。整个函数只调用了 MATLAB 中基础的 fft 和 rectwin 函数，分帧、加窗、频谱整理的逻辑均通过循环和控制语句自主完成。矩形窗的简单性使得代码结构清晰，同时为后续替换窗函数或调整帧移参数保留了灵活性。该函数将在时频图绘制和音乐动画生成中重复调用，是整个时频分析流程的算法核心。在实验中通过绘制扫频信号和已知频率成分信号的时频图，确认了实现的准确性。

(3) 时频图的生成

在获得短时傅里叶变换的复数频谱矩阵之后，需要将其以可视化方式呈现为时频图，从而直观地观察信号频率分量随时间的变化规律。时频图通常采用二维伪彩色图表示，横轴对应时间，纵轴对应频率，颜色的深浅或色调表示该时刻、该频率处的频谱幅值大小。本文采用 MATLAB 中的 imagesc 函数来生成时频图。

首先调用上一节编写的 my_stft 函数，对信号计算得到矩阵 S 以及相应的频率轴 f 和时间轴 t。矩阵 S 的元素是复数，其模值代表该时频点处的幅度。为了在有限的显示动态范围内更清晰地分辨强弱分量，将线性幅度转换为分贝（dB）表示：

$$S_{dB}(i, j) = 20 \cdot \log_{10}(|S(i, j)| + \varepsilon)$$

其中 ε 为一极小正值，以避免对零取对数。进一步地，为使不同窗长下的时频图具有可比性，将所有时频图的色阶范围统一固定为 $[-60, 0]$ dB，即幅度最强的分量对应 0 dB，低于 -60 dB 的部分显示为背景色。这样处理后的数据矩阵直接交由 imagesc 绘制，横纵坐标分别用 t 和 f 标注，并设置 axis xy 使纵轴方向由下至上递增，符合常规频谱图的阅读习惯。颜色映射选用 jet，并附加色标以标明幅值对应关系。

实验中对每组信号均生成了四幅不同窗长（64, 128, 256, 512 点）下的时频图，以子图形式拼接成一幅大图，便于横向比较。绘制时固定帧移 hop = 32 点，确保时间轴的密度一致。图题和坐标轴标签已标注清楚，所有图像均以栅格线辅助读图。

通过上述方式，信号在不同时刻、不同频率上的能量分布一目了然。例如，扫频信号的时频图呈现为一条从低频延伸至高频的斜线，而多音脉冲信号的时频图则表现为若干水平亮线及一个孤立亮块，分别对应持续的正弦分量和瞬态脉冲。这些特征在单纯傅

里叶幅度谱中是无法观察到的，从而凸显了时频图在分析非平稳信号时的优势。该可视化方法将被用于后续窗长对比实验以及音乐动画的分析，作为展现算法效果的主要载体。

(4) 实时音乐频谱动画的实现

在完成合成信号的 STFT 分析与可视化后，本实验进一步将方法应用于实际的音乐信号，实现随音乐播放而实时变化的频谱柱状图动画，效果参考了 Bilibili 网站上的相关视频。该动画的生成完全基于前文所述的自编 STFT 函数，未使用任何外部现成的音频分析库或频谱显示控件。

首先，通过 MATLAB 的 `audioread` 函数读取用户指定的 MP3 音频文件，并将立体声的双声道平均化为单声道信号。为与前两部分实验保持一致，将音频信号重采样至 4410Hz 的采样率。若需限制分析时长（例如仅分析前 30 秒），可对信号进行截取；本实验为完整展现音乐全过程，默认分析整个音频文件。

接着，调用 `my_stft` 函数对整段重采样后的音乐信号进行矩形窗短时傅里叶变换。窗长选为 100 ms（对应 441 个采样点，调整为偶数后使用），帧移设定为窗长的 1/8，以获得高度重叠的时间帧和流畅的动态效果。计算完成后得到复数频谱矩阵 S 、频率向量 f 和时间向量 t 。为兼顾计算效率与视觉效果，截取 20Hz 至 2500Hz 的频率范围，提取对应的幅度谱 $|S|$ 。

为使柱状图动画平稳且美观，将关心的频率区间均匀划分为若干等宽频带（实验中取 96 个频带），对每个频带内所有频率分量的幅度取平均值，得到每个时间帧上各频带的平均幅度。这一处理不仅平滑了频谱，还大幅减少了图形更新的数据量，降低了实时渲染的负担。

实时动画的驱动采用“预计算+播放时同步”的策略。利用 `audioplayer` 对象播放音频，播放开始后，程序进入一个 `while` 循环，持续检测播放器当前所处的采样点位置，将其映射到对应的时间坐标，在预计算的 STFT 时间向量中寻找最近邻帧，读取该帧的各频带幅度，并通过 `bar` 图对象更新柱状图的高度。纵轴采用线性幅度刻度，其上限固定为全部幅度最大值乘以 0.8，使得柱状图在大多数时刻能占据大部分显示区域，避免因短暂尖峰而压缩整体动态范围。横轴显示频率，纵轴显示幅度，标题栏显示当前播放时间。循环间隔通过 `pause` 控制，约为 0.05 秒，对应约 20 帧/秒的刷新率，保证了动画的连续性。

为了获得较好的视觉体验，柱状图的宽度设为 0.3 左右，边缘绘制浅色细线，使相邻柱子间保持微小间隙。同时，在图形窗口的 `CloseRequestFcn` 回调函数中设置了窗口

关闭时自动停止音频播放并释放资源的逻辑。此外，通过逐帧捕获图形窗口图像并写入 VideoWriter 对象，可选择地将实时动画保存为无声的 AVI 视频文件，以便于静态展示或后续的音频合成。

该实时频谱动画充分展示了自编 STFT 在真实音频信号分析中的可用性。音乐播放过程中，低频的鼓点使低频频带柱子显著跃起，而高音旋律则引发高频频带的活动，直观反映了音乐中不同频率成分在时间轴上的能量变化。实验证实，基于矩形窗 STFT 的频谱柱状图动画能够很好地实现音乐信号的时频可视化，达到了参考视频类似效果。

四、实验与结果分析

(1) 两个测试信号的频谱对比

作为整个实验的起点，首先对 3.1 节所构造的两段测试信号计算傅里叶变换，获得其幅度谱，从而直接观察传统傅里叶分析在面对非平稳信号时的表现。

图 1 给出了信号 1 和信号 2 在 0 ~ 2205 Hz 范围内的单边幅度谱。信号 1 为 200 ~ 2000 Hz 的线性扫频信号，其频谱呈现为一个平坦的宽频带，能量几乎均匀地分布在整个扫频区间内，仅在起始频率和终止频率附近出现小幅尖峰，中间频段保持较为平稳的幅度。这一结果是可以预见的：由于扫频信号在 2 秒内遍历了 200 Hz 到 2000 Hz 的所有频率，傅里叶变换是对整段信号进行积分，无法区分不同时刻的频率成分，最终只能给出所有频率在总时长内能量累积的平均表现。

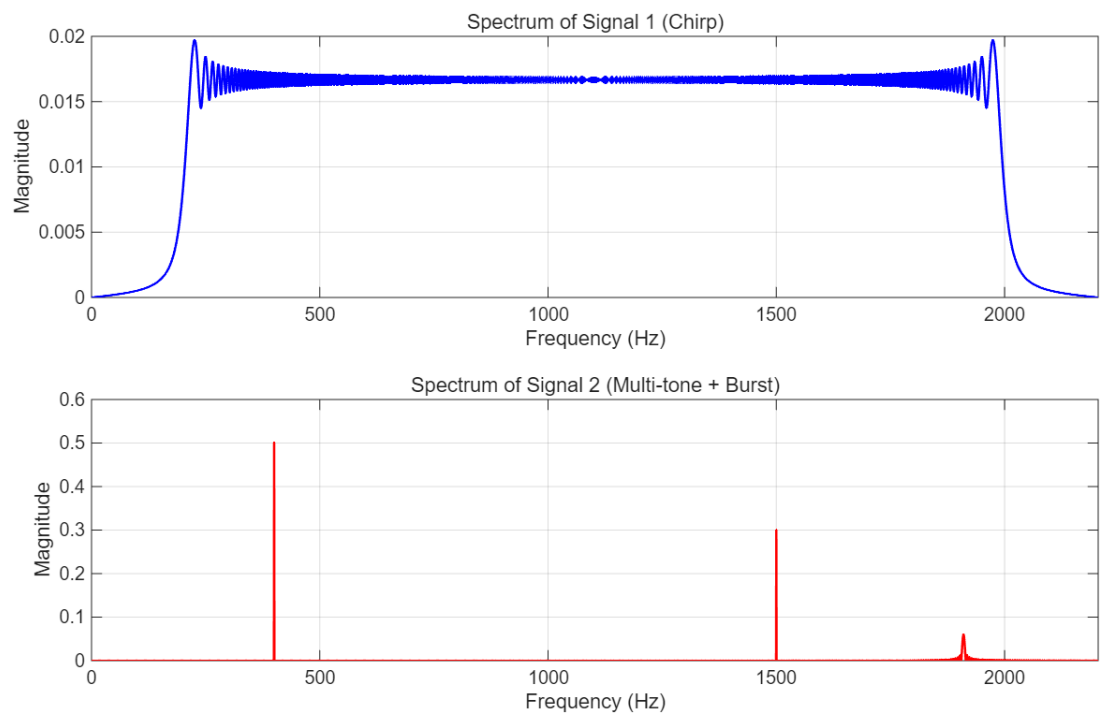


图 1 测试信号的傅里叶幅度谱对比

信号 2 的频谱则呈现出截然不同的特征。在 400Hz 和 1500Hz 处存在两根尖锐而高耸的谱线，分别对应于两个持续存在的正弦分量。此外，在 1900 Hz 附近还出现了一根幅度较低的谱线。值得注意的是，2500 Hz 的瞬态脉冲原本位于奈奎斯特频率 2205Hz 之外，在 4410 Hz 的采样率下会产生频率混叠，其折叠频率为 $4410 - 2500 = 1910\text{Hz}$ ，因此该谱线的出现恰好是混叠的结果。然而，从频谱中完全无法判断该 2500 Hz 分量仅存在于 0.8 ~ 1.0 秒这一短暂的时段内，它与其相邻的持续正弦分量在频谱图上表现为同等地位的单一峰线，时间信息完全丢失。

对比两张频谱图可以发现，单纯通过傅里叶变换无法获得频率成分的时间定位信息。对于信号 1，频谱呈现宽带连续谱，看似所有频率同时存在，但实际上频率是随时间依次出现的；对于信号 2，2500Hz 的瞬态脉冲与持续存在的 400Hz 和 1500 Hz 分量在频谱图上没有本质区别，其短暂的时间特征被完全掩盖。这表明，当信号为非平稳或包含局部事件时，傅里叶变换的全局频谱不能充分反映信号的真实特性，必须引入时频分析手段来弥补这一不足。这也正是下一节使用短时傅里叶变换对信号进行时频联合分析的根本动因。

(2) 不同窗长对时频图的影响

为了考察窗口长度对短时傅里叶变换时频分辨率的影响，利用 `my_stft` 函数，分别计算了两个信号在矩形窗长 $L = 64, 128, 256, 512$ 下的时频图，帧移统一固定为 32 点。色阶范围统一设为 $[-60, 0]\text{dB}$ ，以 dB 值显示幅度，便于跨窗长对比。

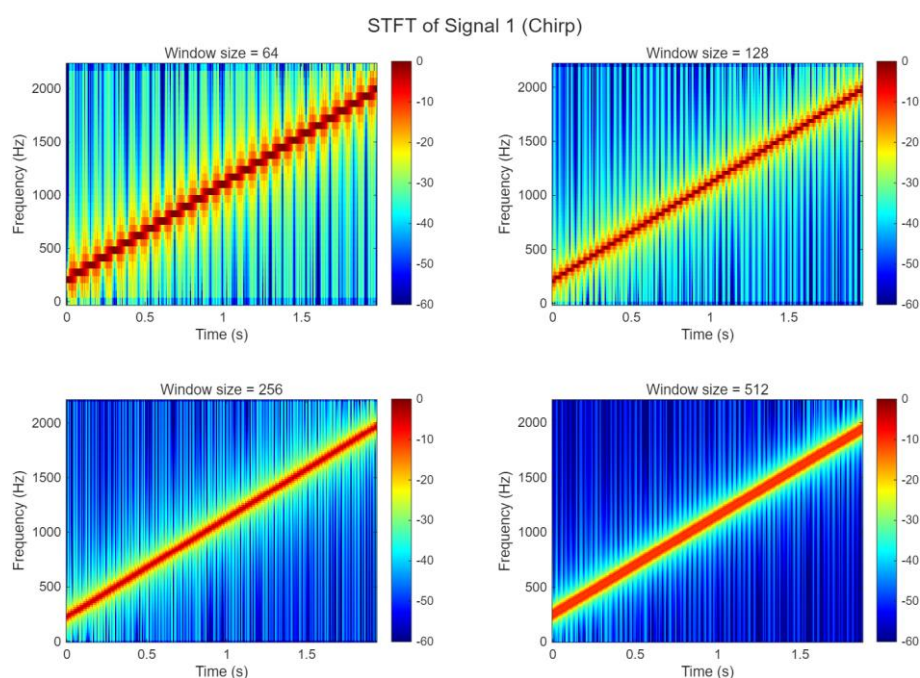


图 2 线性扫频信号的时频图

信号 1 线性扫频信号的时频图

图 2 给出了信号 1 在四种窗长下的时频图。整体上，各子图中均呈现出一条从低频逐渐延伸至高频的斜向亮带，准确地反映了信号频率随时间线性增加的特性，这与信号的 chirp 特性完全吻合。然而，不同窗长下图中的斜线宽窄和边缘清晰度存在显著差异。

当窗长较小 ($L = 64$) 时，斜线在频率方向上的宽度较大，色带由红至黄、青纵向弥散明显，表明频率分辨率较低。同时，斜线的起止点在时间轴上边界清晰，未出现明显的横向拖尾，意味着时间分辨率较高。这是因为短窗口包含的信号点数少，频域的主瓣较宽，但窗口在时域的支撑区短，能够较精确地定位时刻。

随着窗长增大到 128 和 256，斜线的纵向宽度逐渐收窄，黄色和青色的弥散区域缩小，大部分背景变为深蓝，表明频率分辨率得到提升；而斜线的起始端和结束端开始出现轻微的横向扩展，时间定位能力略有下降。

当窗长增大至 512 时，斜线的频率分辨率达到最高，亮带在纵向上变得纤细，但在信号的两端，尤其是扫频起始和结束处，出现了明显的横向模糊，即时间分辨率显著降低。这说明长窗口虽然能够更好地分辨相邻频率，却将窗口内的时间变化平均化了，导致快速变化时刻的定位变得模糊。

上述现象完全符合短时傅里叶变换的不确定性原理：窗长 L 与频率分辨率 $\Delta f = f_s/L$ 成反比，而与时间分辨率 $\Delta t \approx L/f_s$ 成正比。在 chirp 信号这类频率连续变化的信号中，这一矛盾表现得尤为直观。

信号 2 多频混合与瞬态脉冲信号的时频图

图 3 展示了信号 2 在不同窗长下的时频图。图中最显眼的特征是两条贯穿整个时间轴的水平亮带，分别位于约 400 Hz 和 1500 Hz 处，对应两个持续存在的正弦分量。除此之外，在时间轴约 0.8 秒、频率约 1900~2000 Hz 区域，出现了一个孤立的亮块，该亮块正是 2500Hz 瞬态脉冲经采样混叠后在频域的表现，其持续时长与原始脉冲相吻合。

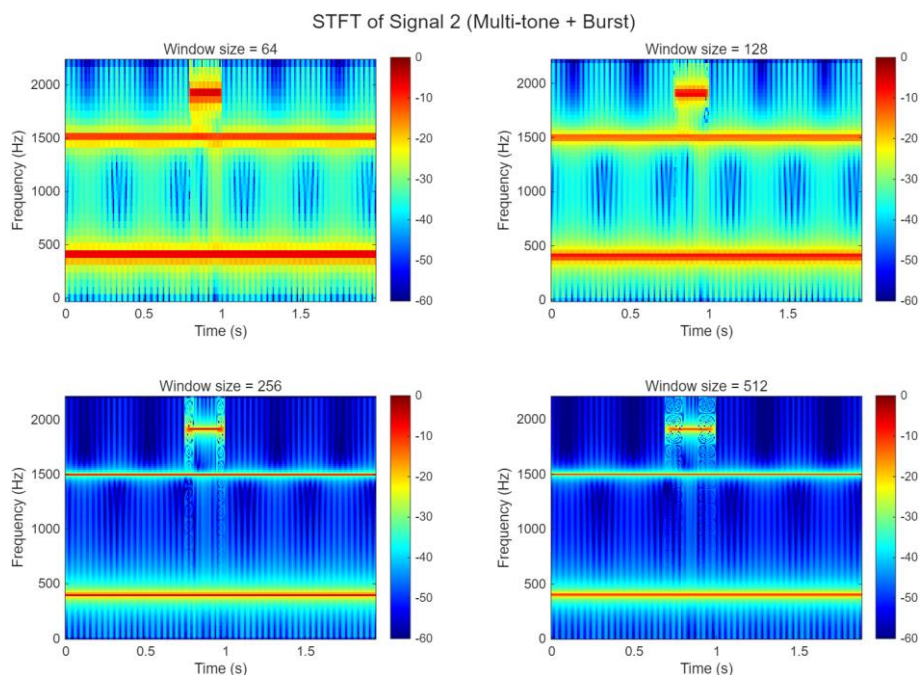


图 3 多频混合与瞬态脉冲信号的时频图

窗长的影响在该图中同样清晰可辨，当 $L = 64$ 时，两条水平亮带纵向较宽，边缘弥散，400 Hz 与 1500 Hz 的谱线显得粗大，频率分辨率欠佳；但对于瞬态脉冲，其在时间轴上的位置非常精确，亮块的横向跨度短，时间定位准确。

当窗长增加至 128 和 256 时，水平亮带逐渐变窄，谱线趋于尖锐，频率分辨率提高；与此同时，瞬态脉冲亮块在时间方向上被逐渐拉长，开始出现横向模糊，时间分辨率下降。

在 $L = 512$ 的极端情况下，两条水平亮带已变得极为纤细，频率定位十分精确；但瞬态脉冲沿时间轴被拉伸到约 0.7~1.0s 的范围，实际脉冲宽度被高估。这意味着长窗口将瞬态事件在时间上涂抹开了，无法再准确反映脉冲的短暂特性。

这一对比生动地揭示了窗长对稳态分量与瞬态分量分析效果的不同影响。对于持续存在的频率分量，长窗有利于精细分辨各频率成分；但对于突发的短脉冲或频率突变，短窗更能准确定位其发生的时刻。

综合以上实验结果，窗长的选择在 STFT 中始终是一个折衷问题，并没有一种最优的窗长能同时满足所有需求。在实际应用中，应当根据信号的具体特征和分析目标来决定窗长：若关心频率的精细结构，宜选用长窗；若侧重于捕捉瞬态事件或频率的快速变化，则短窗更为合适。这一规律在后续对真实音乐信号的时频分析中同样具有指导意义。

(3) 音乐信号的时频动画

为了检验本文实现的矩形窗短时傅里叶变换在真实音频信号中的分析效果,并直观展示音乐中频率成分随时间的变化过程,本节选取了一首 MP3 格式的音乐,利用 `my_stft` 函数提取其时频谱,并制作了随音乐播放同步更新的实时频谱柱状图动画。

实验所选音乐为《星茶会》。音频文件本身的采样率由 MATLAB 的 `audioread` 函数读取获得,为与前述实验保持一致的参数设置,同时降低预计算和实时渲染的计算量,程序首先将音频重采样至 4410 Hz 的采样率,并将双声道向下混合为单声道信号。调用 `my_stft` 函数对整段音乐进行矩形窗 STFT,窗长取 100ms (对应 441 个采样点,经调整为偶数后使用),帧移设为窗长的 1/8,以提供高时间分辨率。计算所得复频谱矩阵的幅度被截取至 20 Hz 至 2500 Hz 的显示区间,以覆盖音乐的主要能量频段。

为得到适合柱状图显示的频带能量,将前述频率区间均匀划分为 96 个等宽频带,对每一帧中落在各频带内的频率幅度求平均,最终得到大小为 $96 \times \text{帧数}$ 的幅度矩阵。纵轴显示幅度采用线性标度,上限设定为全曲幅度最大值的 0.8 倍,使得绝大部分时间段的柱子均能占据足够的视觉高度,同时避免因少数极高幅度尖峰造成整体动态范围的过度压缩。

实时动画的核心逻辑为:首先基于处理后的音频数据创建 `audioplayer` 对象,并开始音频播放。随后进入一个循环,以约 20 Hz 的频率检测当前播放位置,根据播放时间在预计算的 STFT 时间向量中匹配对应的频谱帧,将该帧的各频带幅度赋值给柱状图对象的 `YData` 属性,实现画面的连续更新。柱状图采用宽度 0.3 的直立矩形条,颜色为蓝色调,相邻柱子保留微小间隙,兼顾美观和可辨识度。图形窗口的关闭回调函数被设置为自动停止音频播放并释放资源,确保了程序操作的便捷性。此外,通过逐帧捕获图形窗口的方式,可以选择将整个动画过程保存为无声 AVI 视频文件,便于静态展示和存档。

图 4 从所生成的动画中截取了若干关键时刻的静止画面。从动画中可以清楚地观察到,音乐低频成分(例如鼓声)导致低频区域柱子的显著升高,而高频旋律的出现则引发高频区域柱子的上冲。能量在不同频带之间的迁移与音乐的节奏、旋律高度吻合,反映了音乐信号在时域上快速变化的频率特性。当用户关闭图形窗口时,音频播放立即停止,动画不再更新,实现了完整的资源管理。

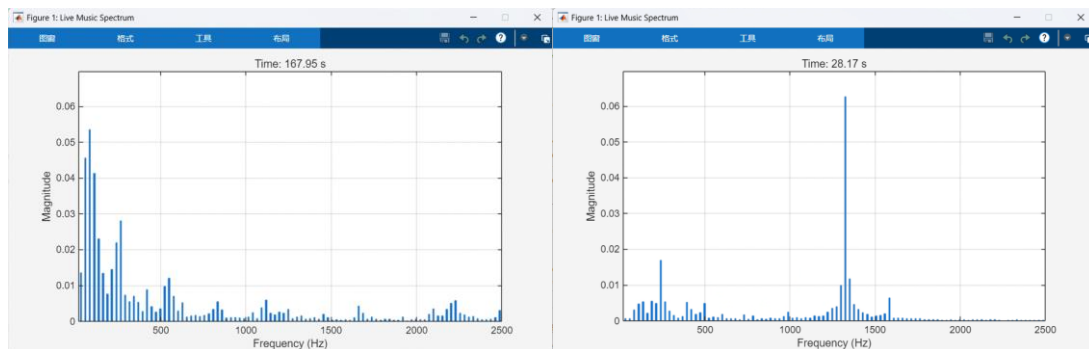


图 4 音乐信号的时频动画

实验表明，基于自编矩形窗 STFT 的实时频谱动画能够生动、有效地展现音乐信号的时频谱变化过程，达到了与参考视频类似的可视化效果。该动画不仅验证了所实现 STFT 算法在真实音频信号上的正确性，也为信号时频分析的教学演示和音乐可视化提供了一种简便的实现方案。若需进一步改善视觉表现，可考虑采用对数幅度刻度、优化窗函数类型或增加频带平滑等方式，这留待后续研究中进行探讨。

五、结论

本文围绕短时傅里叶变换（STFT）这一基本的时频分析工具，从算法实现到实际应用进行了较为完整的探讨和实验。主要工作包括以下三个方面。

首先，通过构造线性扫频信号与多频叠加瞬态脉冲信号，对比了传统傅里叶变换与短时傅里叶变换在分析非平稳信号时的不同表现。全局傅里叶幅度谱只能给出信号在观测时段内频率的平均能量分布，无法揭示频率成分随时间的变化顺序和局部瞬态特征；而 STFT 通过滑动加窗的方式，将一维信号展开为二维的时频谱，清晰地展现了信号频率随时间演化的过程。

其次，自编了基于矩形窗的 STFT 函数，并利用该函数绘制了两个信号在不同窗长下的时频图。实验结果表明，窗长决定了时间分辨率与频率分辨率的折衷关系：短窗有利于捕捉信号的快速瞬变，但频率分辨能力不足；长窗能够精细区分邻近频率分量，却会模糊时间上的局部细节。这一规律对于实际应用中窗长的选择具有重要的指导意义。

最后，将自编 STFT 应用于真实音频信号，制作了随音乐播放而实时跳动的频谱柱状图动画。软件采用预计算与实时刷新相结合的方式，兼顾了计算效率与播放流畅度。动画直观地展示了音乐中不同频段能量随时间的变化，验证了前述算法在真实音频信号上的有效性和实用性。

综上所述，本文通过理论推导、算法实现和实验验证，系统演示了矩形窗短时傅里

叶变换在信号时频分析中的功能与特点。实验表明，基于矩形窗的 STFT 是一种实现简单、效果直观的时频分析方法，能够胜任非平稳信号的可视化分析任务，对于理解傅里叶变换的局限性以及学习时频分析基础具有较好的帮助作用。本文的工作也存在一些可以进一步改进的地方：例如，矩形窗的频谱泄漏较大，可尝试采用汉明窗、高斯窗等窗口函数以改善谱的质量；频域的分辨率和显示方式也有优化空间；实时动画中若能将音频与画面精确同步并输出有声视频，展示效果将更加完整。这些问题有待在后续的研究中继续探索。

参考文献

[1]于慧敏. 信号与系统[M]. 2 版. 北京: 化学工业出版社, 2007.8.

[2]江流不染. matlab 音乐频谱动画[EB/OL]. 2020-03-28.

<https://www.bilibili.com/video/av667556695/>