

信号系统 Project2

Introduction :

1、掌握时域处理方法：

(1) 数字下变频

数字下变频是一种将高频信号在数字域中平移到低频（基带）的技术。它是现代通信、雷达、声纳系统中信号预处理的核心步骤之一，通常用于从采集到的高频信号中提取出有用信息，降低处理的复杂度

(2) 数字滤波器设计及使用

数字滤波器是数字信号处理中最重要的工具之一，用来增强或抑制信号中某些频率成分，从而实现特定频率范围信号的通过、阻断或衰减。设计滤波器需要设计合理的采样率等数据，并验证滤波器是否合适

(2) 互相关计算（模糊函数）

互相关是用来衡量两个信号相似程度的一种方法，尤其是用于找出一个信号在另一个信号中“出现的位置”和“延迟”。

而在雷达和通信系统中，模糊函数是描述发送信号与接收信号在时延和多普勒频移二维空间上的相关性的一个工具

2、掌握频率处理方法

(1) 分析信号频谱

分析信号频谱是数字信号处理中的核心任务之一，揭示了信号在频率域中的能量分布情况。常用于语音识别、雷达通信、图像处理等领域。常用频谱分析方法有离散傅里叶变换，连续傅里叶变换，welch 等

(2) 低通滤波设计及验证

```
% 滤波器设计
clc,clear;
load('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_1.mat');
n = round(0.01 * f_s);
REF = seq_ref(1:n);
SUR = seq_sur(1:n);
t = (0:length(REF)-1)/f_s;
t = t(1:n);

lpFilt = designfilt('lowpassfir', ...
    'PassbandFrequency', 5e6, ...
    'StopbandFrequency', 7e6, ...
    'PassbandRipple', 0.5, ...
    'StopbandAttenuation', 50, ...
    'SampleRate', f_s, ...
    'DesignMethod', 'equiripple');
```

```

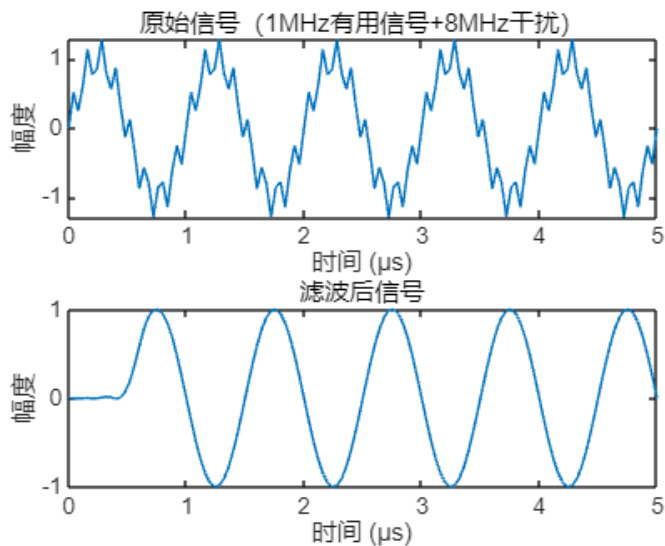
% 验证
test_signal = sin(2*pi*1e6*t) + 0.3*sin(2*pi*8e6*t);

% 滤波处理
filtered_signal = filter(lpFilt, test_signal);

% 时频对比
figure;
subplot(2,1,1);
plot(t*1e6, test_signal);
title('原始信号 (1MHz 有用信号+8MHz 干扰) ');
xlabel('时间 (μs)'); ylabel('幅度');
xlim([0 5]); % 显示前 5μs

subplot(2,1,2);
plot(t*1e6, filtered_signal);
title('滤波后信号');
xlabel('时间 (μs)'); ylabel('幅度');
xlim([0 5]);

```



类型：fir 低通滤波器

通带截止频率：5 MHz 以下的信号会保留

阻带起始频率：7 MHz 以上的信号会大幅衰减

通带波动：最大 0.5 dB

阻带衰减：最少 50 dB，表示强抑制

采样率 f_s ：你之前定义的是 $f_s = 25000000\text{Hz}$

验证：

原始信号：有明显的快速震荡，低频包络在高频振荡上呈现，有“毛刺”

滤波后的信号：变得光滑，振荡消失，波形接近纯正弦（只有 1 MHz 频率）

结论：滤波器成功保留 1 MHz，抑制了 8 MHz 成分，滤波器成功作用

3、 基于通信信号的运动检测技术在生活中的应用

随着科技的快速发展，基于通信信号的运动检测技术已经成为现代生活中不可或缺的一部分。这项技术利用信号处理的原理，通过分析接收到的信号变化来检测和跟踪目标的运动状态。它在多个领域有着广泛的应用，包括但不限于以下几个方面：

1. 雷达系统

在军事和民用雷达系统中，运动检测技术用于监测和跟踪空中、海上和地面的目标。它可以帮助雷达系统识别和分类目标，如飞机、船只和车辆，从而为决策提供关键信息。

2. 无线通信

在无线通信领域，运动检测技术可以用于优化信号传输。例如，在移动通信中，通过检测用户的移动状态，基站可以调整信号的传输参数，以提高通信质量和效率。

3. 智能交通系统

智能交通系统利用运动检测技术来监控交通流量，预测交通拥堵，并指导车辆避开拥堵区域。此外，它还可以用于检测交通事故和异常行为，提高道路安全。

4. 健康监测

在医疗健康领域，运动检测技术可以用于监测患者的活动水平和行为模式。例如，可穿戴设备可以利用这项技术来追踪用户的运动数据，帮助他们更好地管理健康。

Tasks:

```
%task1
clc,clear;
%导入数据
Duration = 0;
REF = [];
SUR = [];
fs = 0;
fc = 0;
N = 0;

for k = 1:20
    fname = sprintf('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_%d.mat', k);
    load(fname);
    n = round(0.01 * f_s);
    N = N+n;
    seq_ref = seq_ref(1:n);
    seq_sur = seq_sur(1:n);
    REF = [REF, seq_ref];
    SUR = [SUR, seq_sur];
```

```

    Duration = Duration + duration;
    fs = fs + f_s;
    fc = fc + f_c;
end

fs = fs/20;
fc = fc/20;
% 时间轴
t = (0:length(REF)-1)/fs;
t = t(1:N);

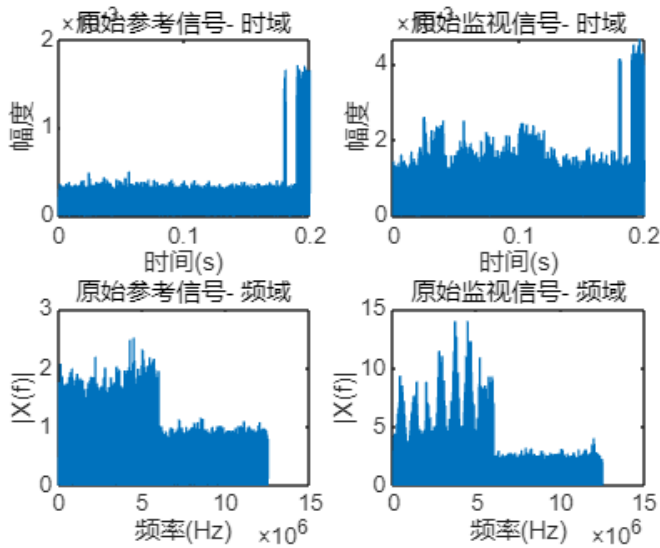
% original
figure;
subplot(2,2,1);
plot(t, abs(REF));
title('原始参考信号- 时域');
xlabel('时间(s)'); ylabel('幅度');

subplot(2,2,2);
plot(t, abs(SUR));
title('原始监视信号- 时域');
xlabel('时间(s)'); ylabel('幅度');

% 频谱（取单边频谱）
f = (0:N-1)*(fs/N);
subplot(2,2,3);
X_ref = fft(REF);
plot(f(1:N/2), abs(X_ref(1:N/2)));
title('原始参考信号- 频域');
xlabel('频率(Hz)'); ylabel('|X(f)|');

subplot(2,2,4);
X_sur = fft(SUR);
plot(f(1:N/2), abs(X_sur(1:N/2)));
title('原始监视信号- 频域');
xlabel('频率(Hz)'); ylabel('|X(f)|');

```



```
clear X_ref;
clear X_sur;
```

对于结果的解释：展示了每个信号的时域波形（原始波形对时间的变化），频域波形（经过 fft 转换后的频率成分），并对 20 个信号取 0.01s 合并在一起。

```
%frequency conversion
cos_f = cos(2*pi*fc*t);

ref_down = REF .* cos_f;
sur_down = SUR .* cos_f;

figure;
subplot(2,2,1);
plot(t, real(ref_down));
title('参考信号 (after) - 时域');
xlabel('时间(s)'); ylabel('幅度');

subplot(2,2,2);
plot(t, real(sur_down));
title('监视信号 (after) - 时域');
xlabel('时间(s)'); ylabel('幅度');

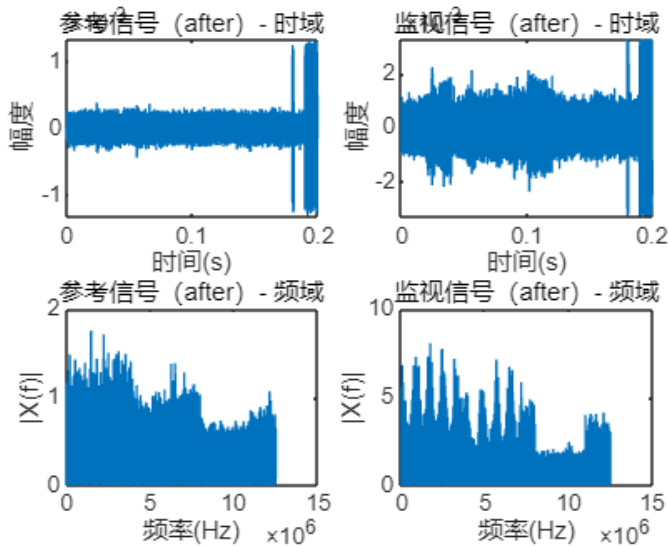
% 频谱
subplot(2,2,3);
X_ref_down = fft(ref_down);
plot(f(1:N/2), abs(X_ref_down(1:N/2)));
title('参考信号 (after) - 频域');
xlabel('频率(Hz)'); ylabel('|X(f)|');

subplot(2,2,4);
```

```

X_sur_down = fft(sur_down);
plot(f(1:N/2), abs(X_sur_down(1:N/2)));
title('监视信号 (after) - 频域');
xlabel('频率(Hz)'); ylabel('|X(f)|');

```



对于结果的解释和解释：用余弦信号 $\cos(2\pi f_{ct})$ 对原始高频信号进行了下变频得到了两个新的信号 `ref_down` 和 `sur_down`，并画出了他们的时域波形和频域波形。

```

% low-pass filtering
% 滤波器
lpFilt = designfilt('lowpassfir', ...
    'PassbandFrequency', 5e6, ...
    'StopbandFrequency', 7e6, ...
    'PassbandRipple', 0.5, ...
    'StopbandAttenuation', 50, ...
    'SampleRate', fs, ...
    'DesignMethod', 'equiripple');

% 应用滤波器
ref_filt = filter(lpFilt, ref_down);
sur_filt = filter(lpFilt, sur_down);

figure;
subplot(2,2,1);
plot(t, real(ref_filt));
title('参考信号 (低通滤波) - 时域');
xlabel('时间(s)'); ylabel('幅度');

subplot(2,2,2);
plot(t, real(sur_filt));
title('监视信号 (低通滤波) - 时域');
xlabel('时间(s)'); ylabel('幅度');

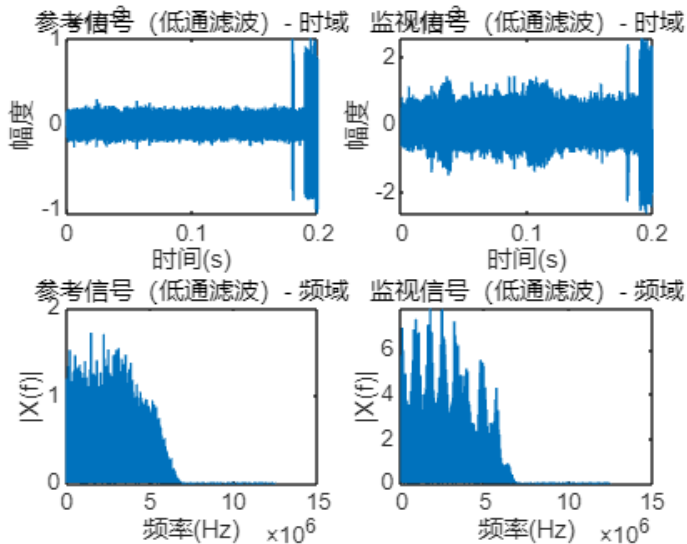
```

```

% 频谱
subplot(2,2,3);
X_ref_filt=fft(ref_filt);
plot(f(1:N/2), abs(X_ref_filt(1:N/2)));
title('参考信号 (低通滤波) - 频域');
xlabel('频率(Hz)'); ylabel('|X(f)|');

subplot(2,2,4);
X_sur_filt=fft(sur_filt);
plot(f(1:N/2), abs(X_sur_filt(1:N/2)));
title('监视信号 (低通滤波) - 频域');
xlabel('频率(Hz)'); ylabel('|X(f)|');

```



对于结果的解释：滤波后的频谱相较于 ref_down/sur_down 明显变窄，原先在较高频率的成分已被显著削弱；大部分的频率能量集中在较低频段（0~5 MHz），符合滤波器设计参数。从频谱来看，滤波器在 5-7MHz 之间成功抑制了能量泄露，有效保留了目标信号核心频率范围，使信号频谱集中，噪声低、干扰小。

```

% % % % % 任务 2 % % % % %
clc,clear;
load("C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_1.mat"); % 0-0.5s 的信号的范围多普勒频谱

% 计算索引
idx_start = round(cur_time * f_s) + 1;
idx_end = round((cur_time + duration) * f_s);

% 防止越界
idx_start = max(1, idx_start);
idx_end = min(size(seq_ref, 1), idx_end);

% 提取当前时间段数据
data_ref = seq_ref(idx_start:idx_end, :);

```

```

data_sur = seq_sur(idx_start:idx_end, :);

% 展开为一维
data_ref_1d = data_ref(:);
data_sur_1d = data_sur(:);

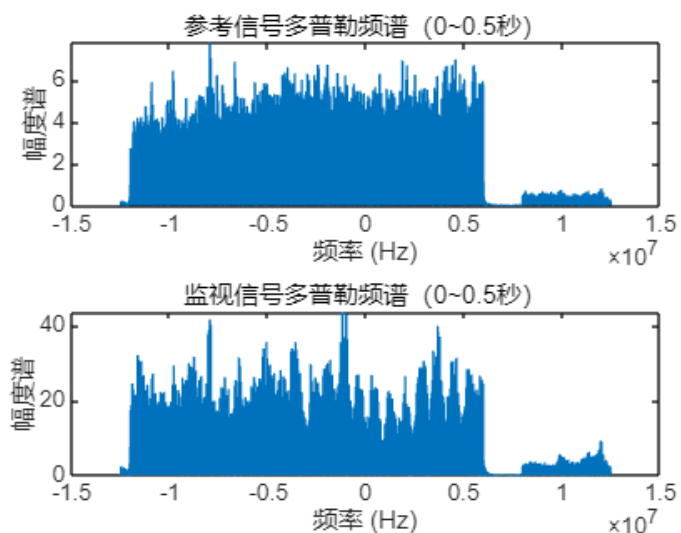
% FFT
NFFT = 2^nextpow2(length(data_ref_1d));
spectrum_ref = abs(fftshift(fft(data_ref_1d, NFFT)));
spectrum_sur = abs(fftshift(fft(data_sur_1d, NFFT)));

% 多普勒频率轴
freq_axis = (-NFFT/2:NFFT/2-1) * (f_s / NFFT);

% 绘图
figure;
subplot(2,1,1);
plot(freq_axis, spectrum_ref);
title('参考信号多普勒频谱 (0~0.5 秒)');
xlabel('频率 (Hz)');
ylabel('幅度谱');

subplot(2,1,2);
plot(freq_axis, spectrum_sur);
title('监视信号多普勒频谱 (0~0.5 秒)');
xlabel('频率 (Hz)');
ylabel('幅度谱');

```



对结果的分析 and 解释：在 0 到 0.5 秒的时间段内，参考信号和监视信号的多普勒频谱显示出相似的频率分布特征。参考信号的幅度谱在 0 Hz 附近有明显的峰值，表明信号中存在较强的直流分量或低频成分。监视信号的幅度谱在 0 Hz 附近也有显著的峰值，但其幅度明显高于参考信号，这可能表明监视信号在该时间段内受到了更强的多普勒效应影响。


```

clc,clear;
load('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_5.mat'); % 2-2.5s 的信号的范围多普勒频谱

% 计算索引
idx_start = round((cur_time-2) * f_s) + 1;
idx_end = round(((cur_time-2) + duration) * f_s);

% 防止越界
idx_start = max(1, idx_start);
idx_end = min(size(seq_ref, 1), idx_end);

% 提取当前时间段数据
data_ref = seq_ref(idx_start:idx_end, :);
data_sur = seq_sur(idx_start:idx_end, :);

% 展开为一维
data_ref_1d = data_ref(:);
data_sur_1d = data_sur(:);

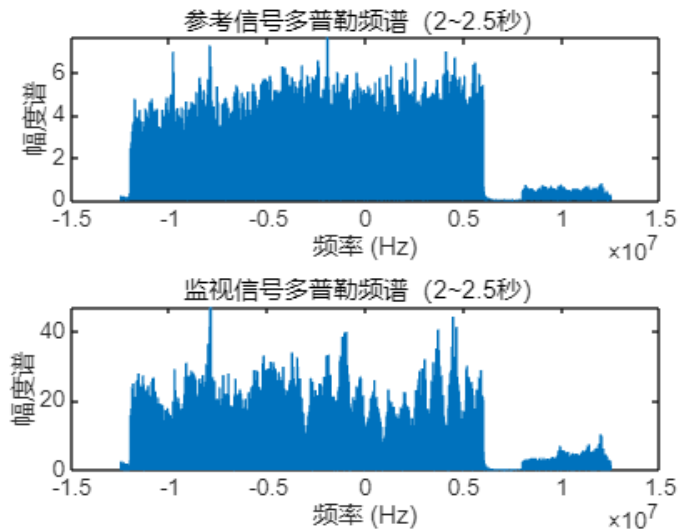
% FFT
NFFT = 2^nextpow2(length(data_ref_1d));
spectrum_ref = abs(fftshift(fft(data_ref_1d, NFFT)));
spectrum_sur = abs(fftshift(fft(data_sur_1d, NFFT)));

% 多普勒频率轴
freq_axis = (-NFFT/2:NFFT/2-1) * (f_s / NFFT);

% 绘图
figure;
subplot(2,1,1);
plot(freq_axis, spectrum_ref);
title('参考信号多普勒频谱 (2~2.5 秒) ');
xlabel('频率 (Hz)');
ylabel('幅度谱');

subplot(2,1,2);
plot(freq_axis, spectrum_sur);
title('监视信号多普勒频谱 (2~2.5 秒) ');
xlabel('频率 (Hz)');
ylabel('幅度谱');

```



对结果的解释：在 2 到 2.5 秒的时间段内，参考信号的多普勒频谱在 0 Hz 附近保持了较高的幅度，但相比 0 到 0.5 秒的时间段，其幅度略有下降。监视信号在这一时间段内的幅度谱显著增加，尤其是在 0 Hz 附近，这可能意味着在这一时间段内，监视信号受到了更强烈的多普勒效应影响，或者信号的动态变化更为剧烈。

```
clc,clear;
load('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_11.mat'); % 5-5.5s 的信号的范围多普勒频谱

% 计算索引
idx_start = round((cur_time-5) * f_s) + 1;
idx_end = round(((cur_time-5) + duration) * f_s);

% 防止越界
idx_start = max(1, idx_start);
idx_end = min(size(seq_ref, 1), idx_end);

% 提取当前时间段数据
data_ref = seq_ref(idx_start:idx_end, :);
data_sur = seq_sur(idx_start:idx_end, :);

% 展开为一维
data_ref_1d = data_ref(:);
data_sur_1d = data_sur(:);

% FFT
NFFT = 2^nextpow2(length(data_ref_1d));
spectrum_ref = abs(fftshift(fft(data_ref_1d, NFFT)));
spectrum_sur = abs(fftshift(fft(data_sur_1d, NFFT)));

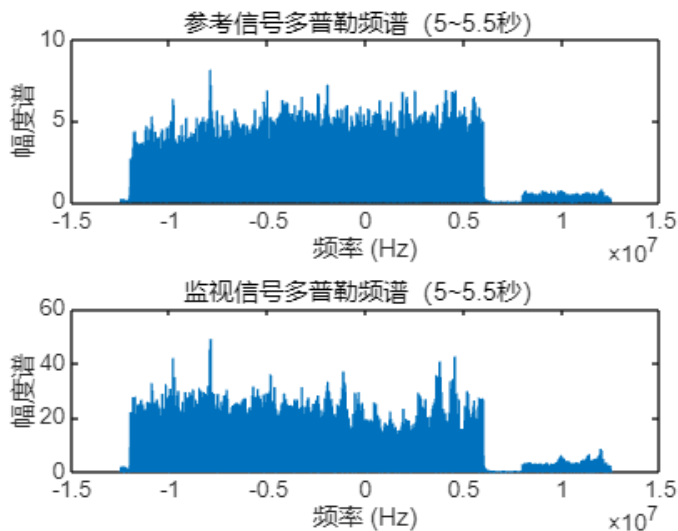
% 多普勒频率轴
freq_axis = (-NFFT/2:NFFT/2-1) * (f_s / NFFT);
```

```

% 绘图
figure;
subplot(2,1,1);
plot(freq_axis, spectrum_ref);
title('参考信号多普勒频谱 (5~5.5 秒) ');
xlabel('频率 (Hz)');
ylabel('幅度谱');

subplot(2,1,2);
plot(freq_axis, spectrum_sur);
title('监视信号多普勒频谱 (5~5.5 秒) ');
xlabel('频率 (Hz)');
ylabel('幅度谱');

```



对结果的解释：在 5 到 5.5 秒的时间段内，参考信号的多普勒频谱在 0 Hz 附近的幅度有所增加，显示出信号的稳定性。监视信号的幅度谱在这一时间段内显著高于参考信号，尤其是在 0 Hz 附近，这可能表明监视信号在这一时间段内受到了更强的多普勒效应影响，或者信号的动态变化更为复杂。

```

clc,clear;
load('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_15.mat'); % 7-7.5s 的信号的范围多普勒频谱

% 计算索引
idx_start = round((cur_time-7) * f_s) + 1;
idx_end = round(((cur_time-7) + duration) * f_s);

% 防止越界
idx_start = max(1, idx_start);
idx_end = min(size(seq_ref, 1), idx_end);

% 提取当前时间段数据
data_ref = seq_ref(idx_start:idx_end, :);
data_sur = seq_sur(idx_start:idx_end, :);

```

```

% 展开为一维
data_ref_1d = data_ref(:);
data_sur_1d = data_sur(:);

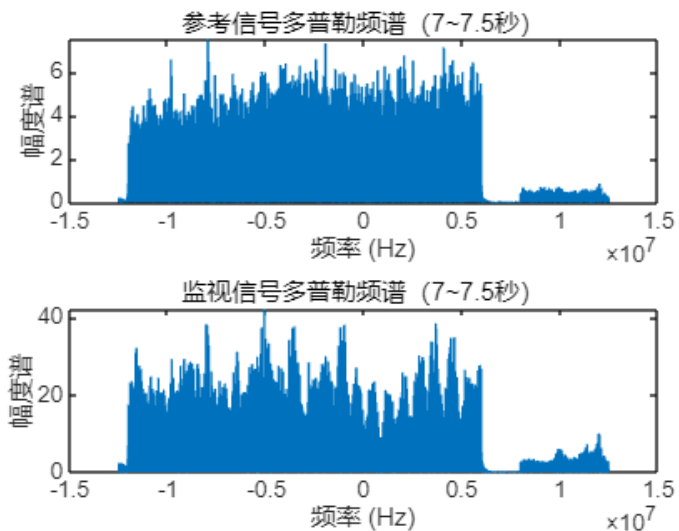
% FFT
NFFT = 2^nextpow2(length(data_ref_1d));
spectrum_ref = abs(fftshift(fft(data_ref_1d, NFFT)));
spectrum_sur = abs(fftshift(fft(data_sur_1d, NFFT)));

% 多普勒频率轴
freq_axis = (-NFFT/2:NFFT/2-1) * (f_s / NFFT);

% 绘图
figure;
subplot(2,1,1);
plot(freq_axis, spectrum_ref);
title('参考信号多普勒频谱 (7~7.5 秒) ');
xlabel('频率 (Hz)');
ylabel('幅度谱');

subplot(2,1,2);
plot(freq_axis, spectrum_sur);
title('监视信号多普勒频谱 (7~7.5 秒) ');
xlabel('频率 (Hz)');
ylabel('幅度谱');

```



对结果的解释和解释：在 7 到 7.5 秒的时间段内，参考信号的多普勒频谱在 0 Hz 附近的幅度保持稳定，而监视信号的幅度谱在这一时间段内略有下降，但仍高于参考信号。这可能表明监视信号在这一时间段内受到了一定程度的多普勒效应影响，但影响程度较之前有所减弱。

综合以上分析，可以得出以下结论：

1. 监视信号在所有时间段内的多普勒频谱幅度均高于参考信号，这可能表明监视信号受到了更强的多普勒效应影响。
2. 在 2 到 2.5 秒和 5 到 5.5 秒的时间段内，监视信号的幅度谱显著增加，这可能意味着在这两个时间段内，监视信号的动态变化更为剧烈。
3. 参考信号在所有时间段内的多普勒频谱幅度相对稳定，显示出信号的稳定性。
4. 监视信号在 7 到 7.5 秒的时间段内受到了一定程度的多普勒效应影响，但影响程度较之前有所减弱。

```
% % % % % 任务 3 % % % % %
N_pulse = 20;
seq_length = length(load('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\
\data_1.mat').seq_ref); % 假设所有长度一致
seq_ref_all = zeros(N_pulse, seq_length);
cur_time_all = zeros(1, N_pulse);

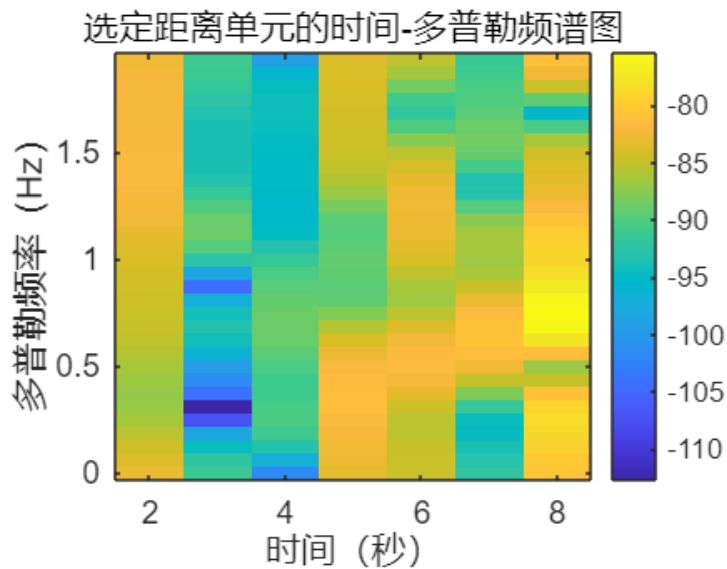
for i = 1:N_pulse
    filename = sprintf('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\
\data_%d.mat', i);
    data = load(filename);
    seq_ref_all(i, :) = data.seq_ref;
    cur_time_all(i) = data.cur_time;
end

range_bin = 100; % 距离单元索引
seq = seq_ref_all(:, range_bin); % 取出慢时间序列

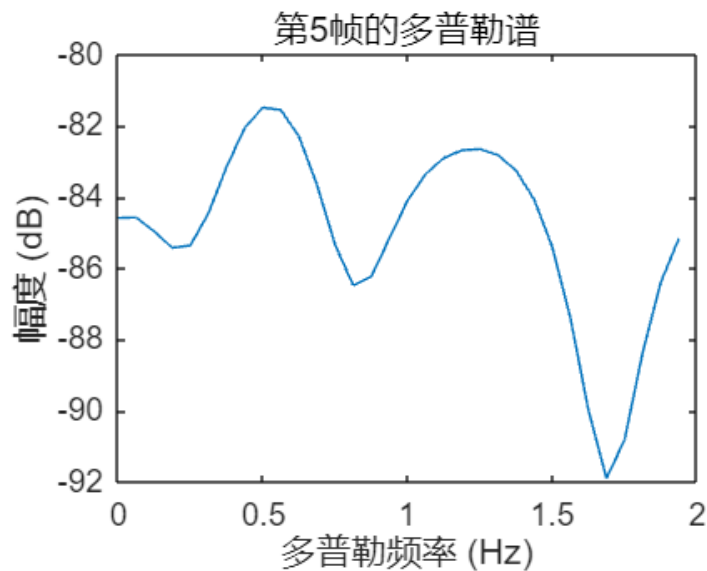
% 计算实际帧率 (PRF) , 假设 cur_time_all 为秒
if N_pulse > 1
    PRF = 1 / mean(diff(cur_time_all));
else
    PRF = 1;
end

win_len = 8;
overlap = round(win_len*0.8);
nfft = 32;
[S,F,T] = spectrogram(seq, win_len, overlap, nfft, PRF);

figure;
imagesc(T+cur_time_all(1), F, 20*log10(abs(S)));
axis xy;
xlabel('时间 (秒) ');
ylabel('多普勒频率 (Hz) ');
title('选定距离单元的时间-多普勒频谱图');
colorbar;
set(gca, 'FontSize', 12);
```



```
% 一维切片：某一帧的多普勒谱
idx = 5; % 比如第 5 帧
figure;
plot(F, 20*log10(abs(S(:, idx))));
xlabel('多普勒频率 (Hz)');
ylabel('幅度 (dB)');
title(sprintf('第%d 帧的多普勒谱', idx));
set(gca, 'FontSize', 12);
```



```
fprintf('第 5 帧对应的时间是: %.3f 秒\n', T(idx) + cur_time_all(1));
```

第 5 帧对应的时间是: 6.000 秒

分析与解释：

1. 拼接慢时间序列

代码首先将 20 个 .mat 文件中的 seq_ref 数据按行排列，形成一个矩阵 seq_ref_all。每一行代表一帧（或一次采样）。然后，选定一个特定的距离单元（例如第 100 个采样点），从 seq_ref_all 中提取该点的幅度，形成慢时间序列 seq。这一步骤的目的是为了分析特定距离单元随时间变化的信号特性。

2. 计算实际帧率

通过计算 cur_time_all 数组中时间差的均值的倒数，得到实际帧率（PRF）。PRF 是多普勒频谱分析中的关键参数，它决定了多普勒频率轴的物理意义。如果 N_pulse 大于 1，即有多个时间点，PRF 通过时间差的平均值得到；如果只有一个时间点，则 PRF 默认为 1。

3. 绘制时间-多普勒谱

使用 spectrogram 函数对慢时间序列 seq 进行短时傅里叶变换，生成时间-多普勒频谱图。图中横轴表示时间（秒），纵轴表示多普勒频率（Hz），色彩表示信号强度（以 dB 为单位）。这个图展示了选定距离单元在不同时间点的多普勒频率分布和强度变化，有助于分析目标的运动特性。

4. 一维谱切片

选定某一帧（例如第 5 帧），绘制该帧的多普勒谱线图，展示多普勒频率与信号幅度的关系。这个图有助于更详细地分析特定时间点的多普勒频率分布，从而推断目标在该时刻的运动状态。

结论：

- 时间-多普勒频谱图（第一张图）显示了选定距离单元在不同时间点的多普勒频率分布和强度变化。图中颜色的变化反映了信号强度的变化，颜色越亮表示信号强度越高。从图中可以看出，多普勒频率在不同时间点有显著变化，这可能与目标的运动状态有关。
- 第 5 帧的多普勒谱（第二张图）展示了第 5 帧在不同多普勒频率下的信号幅度。图中曲线的波动反映了该帧在不同多普勒频率下的信号强度变化。通过分析这些变化，可以推断目标在该时刻的运动特性。

Advance:

加分项 1：频率转换部分用复数混频代替实数混频，去除镜像，频谱更集中，更易滤波，后续可直接提取相位/包络（比如测距）；FFT 前加窗函数，提升频谱可读性，抑制频谱泄漏，减少假频率成分；改用 filtfilt()，双向滤波，消除相位延迟，保证后续处理不受波形偏移影响。

```
% frequency conversion (改为复数下变频)
mix = exp(-1j*2*pi*fc*t);

ref_down = REF .* mix;
sur_down = SUR .* mix;

% 加窗
window = hamming(length(t))';
ref_win = ref_down .* window;
X_ref_down = fft(ref_win);
sur_win = sur_down .* window;
X_sur_down = fft(sur_win);

% 绘图
figure;
```

```

subplot(2,2,1);
plot(t, real(ref_down));
title('参考信号 (复数下变频) - 时域');
xlabel('时间(s)'); ylabel('幅度');

subplot(2,2,2);
plot(t, real(sur_down));
title('监视信号 (复数下变频) - 时域');
xlabel('时间(s)'); ylabel('幅度');

subplot(2,2,3);
plot(f(1:N/2), abs(X_ref_down(1:N/2)));
title('参考信号 (复数下变频) - 频域');
xlabel('频率(Hz)'); ylabel('幅度(dB)');

subplot(2,2,4);
plot(f(1:N/2), abs(X_sur_down(1:N/2)));
title('监视信号 (复数下变频) - 频域');
xlabel('频率(Hz)'); ylabel('幅度(dB)');

% 低通滤波 (使用 firlfilt)
lpFilt = designfilt('lowpassfir', ...
    'PassbandFrequency', 5e6, ...
    'StopbandFrequency', 7e6, ...
    'PassbandRipple', 0.5, ...
    'StopbandAttenuation', 50, ...
    'SampleRate', fs, ...
    'DesignMethod', 'equiripple');

% 使用 firlfilt (双向滤波)
ref_filt = firlfilt(lpFilt, real(ref_down));
sur_filt = firlfilt(lpFilt, real(sur_down));
ref_filt_win = ref_filt .* window;
sur_filt_win = sur_filt .* window;
X_ref_filt = fft(ref_filt_win);
X_sur_filt = fft(sur_filt_win);

% 画图
figure;
subplot(2,2,1);
plot(t, ref_filt);
title('参考信号 (低通滤波) - 时域');
xlabel('时间(s)'); ylabel('幅度');

subplot(2,2,2);
plot(t, sur_filt);
title('监视信号 (低通滤波) - 时域');
xlabel('时间(s)'); ylabel('幅度');

subplot(2,2,3);

```



```

plot(f(1:N/2), abs(X_ref_filt(1:N/2)));
title('参考信号（低通滤波）- 频域');
xlabel('频率(Hz)'); ylabel('幅度(dB)');

subplot(2,2,4);
plot(f(1:N/2), abs(X_sur_filt(1:N/2)));
title('监视信号（低通滤波）- 频域');
xlabel('频率(Hz)'); ylabel('幅度(dB)');

```

综合以上分析，可以得出以下结论：

- 1.在未处理前，目标信息和噪声混杂在一起，不易提取有用成分
- 2.通过下变频变换使得频谱中的主能量区域发生了平移，平移到 x 轴，利于处理，但是 u 时域波形仍有高频成分，与要进行进一步的处理
- 3.使用低通滤波器保留了目标信号，抑制了下变频后的高频副带，信号频谱变得更集中、干净，噪声成分显著减少，时域波形更加平滑。

加分项 2：时间多普勒图在时间刻度上的进一步细分

时间-多普勒图（Time-Doppler Map）是雷达信号处理中的一个重要工具，它能够展示目标在不同时间点上的多普勒频率分布，从而帮助分析目标的运动特性。为了提高时间分辨率，可以采用更小的滑动窗口步长进行短时傅里叶变换（STFT）。

更高时间分辨率的 STFT（滑动窗口步长更小）

spectrogram 的时间分辨率由窗口长度 win_len 和重叠 overlap 决定。

- 步长 = win_len - overlap（帧的移动步长）
- 步长越小，时间轴刻度越细致！

STFT 的时间分辨率由窗口长度（win_len）和重叠（overlap）决定。步长计算公式为步长 = win_len - overlap。步长越小，时间轴刻度越细致。例如，当步长为 1 时，即每次滑动窗口只移动一个样本点，可以得到更高的时间分辨率。

在代码中，通过设置 overlap = win_len - 1，实现了步长为 1 的 STFT，从而提高了时间分辨率。此外，还通过对比不同重叠值（overlap=4 和 overlap=7）的效果，展示了时间分辨率的变化。

```

clc; clear;
N_pulse = 20;
seq_length = length(load('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_1.mat').seq_ref);
seq_ref_all = zeros(N_pulse, seq_length);
cur_time_all = zeros(1, N_pulse);
for i = 1:N_pulse
    filename = sprintf('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\data_%d.mat', i);
    data = load(filename);

```

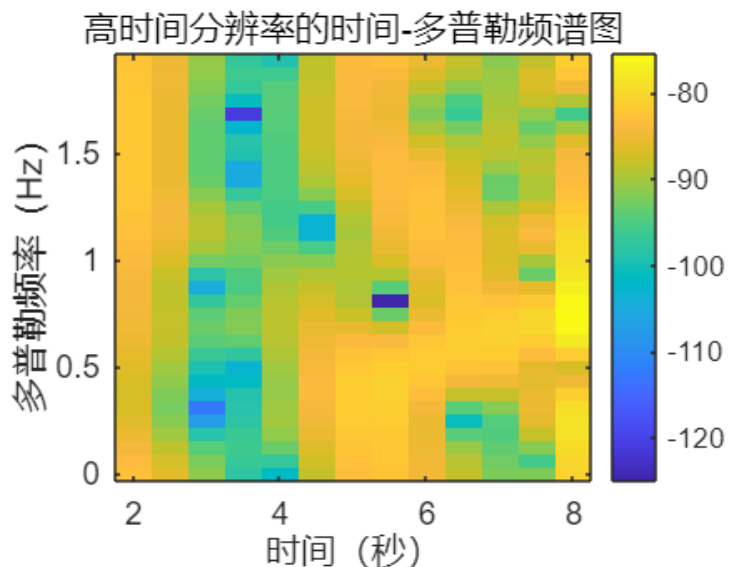
```

    seq_ref_all(i, :) = data.seq_ref;
    cur_time_all(i) = data.cur_time;
end
range_bin = 100;
seq = seq_ref_all(:, range_bin);

if N_pulse > 1
    PRF = 1/mean(diff(cur_time_all));
else
    PRF = 1;
end
win_len = 8;
overlap = win_len - 1;    % 步长 1
nfft = 32;
[S, F, T] = spectrogram(seq, win_len, overlap, nfft, PRF);

figure;
imagesc(T+cur_time_all(1), F, 20*log10(abs(S)));
axis xy;
xlabel('时间 (秒) ');
ylabel('多普勒频率 (Hz) ');
title('高时间分辨率的时间-多普勒频谱图');
colorbar;
set(gca, 'FontSize', 12);

```



```

% ----- 分辨率对比图 -----
figure;
subplot(1,2,1);
win_len = 8; overlap = 4;
[S1, F1, T1] = spectrogram(seq, win_len, overlap, nfft, PRF);
imagesc(T1+cur_time_all(1), F1, 20*log10(abs(S1)));
axis xy; title('overlap=4');
xlabel('时间 (秒) '); ylabel('多普勒频率 (Hz) ');

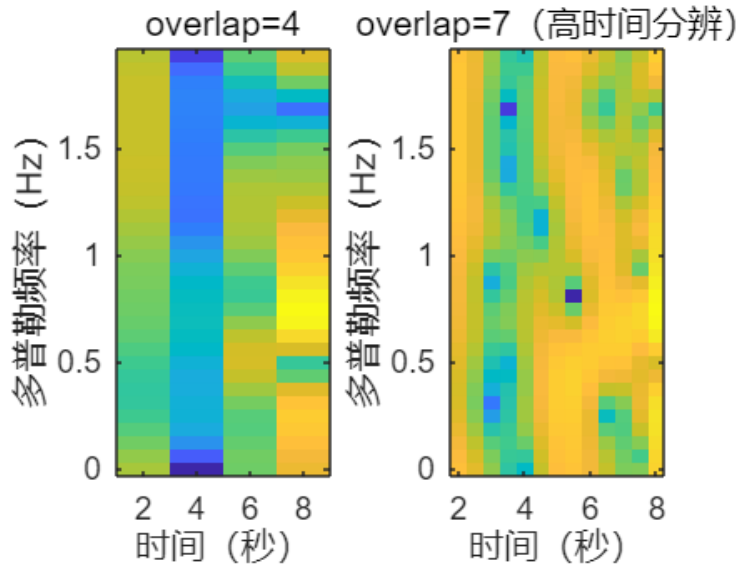
```

```

set(gca, 'FontSize', 12);

subplot(1,2,2);
win_len = 8; overlap = 7;
[S2, F2, T2] = spectrogram(seq, win_len, overlap, nfft, PRF);
imagesc(T2+cur_time_all(1), F2, 20*log10(abs(S2)));
axis xy; title('overlap=7 (高时间分辨) ');
xlabel('时间 (秒) '); ylabel('多普勒频率 (Hz) ');
set(gca, 'FontSize', 12);

```



用插值方法对时间轴细分

样条插值是一种常用的插值技术，它可以在保持数据平滑性的同时，对时间轴进行细分。在代码中，通过将时间轴细分 10 倍，并使用样条插值对每个频率分量进行插值，实现了时间轴的细分。

插值后的时间-多普勒谱图显示了更细致的时间刻度，这有助于更精确地分析目标的运动特性。此外，通过生成插值后多普勒谱随时间变化的动画，可以直观地观察目标运动状态随时间的变化。

```

clc; clear;
N_pulse = 20;
seq_length = length(load('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\
\data_1.mat').seq_ref);
seq_ref_all = zeros(N_pulse, seq_length);
cur_time_all = zeros(1, N_pulse);
for i = 1:N_pulse
    filename = sprintf('C:\\Users\\eliri\\OneDrive\\桌面\\project2\\data\\
\data_%d.mat', i);
    data = load(filename);
    seq_ref_all(i, :) = data.seq_ref;
    cur_time_all(i) = data.cur_time;
end

range_bin = 100;
seq = seq_ref_all(:, range_bin);

```

```

if N_pulse > 1
    PRF = 1/mean(diff(cur_time_all));
else
    PRF = 1;
end

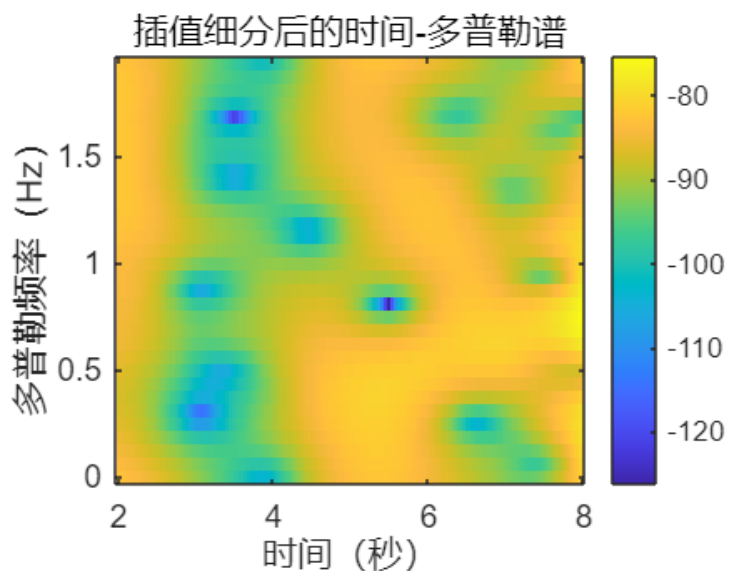
win_len = 8;
overlap = win_len - 1;
nfft = 32;
[S, F, T] = spectrogram(seq, win_len, overlap, nfft, PRF);

T_fine = linspace(T(1), T(end), length(T)*10); % 时间轴细分 10 倍

% 对每个频率分量做插值
S_abs = abs(S);
S_fine = zeros(size(S_abs,1), length(T_fine));
for k = 1:size(S_abs,1)
    S_fine(k,:) = interp1(T, S_abs(k,:), T_fine, 'spline');
end

figure;
imagesc(T_fine+cur_time_all(1), F, 20*log10(S_fine));
axis xy;
xlabel('时间 (秒) ');
ylabel('多普勒频率 (Hz) ');
title('插值细分后的时间-多普勒谱');
colorbar;
set(gca, 'FontSize', 12);

```



```

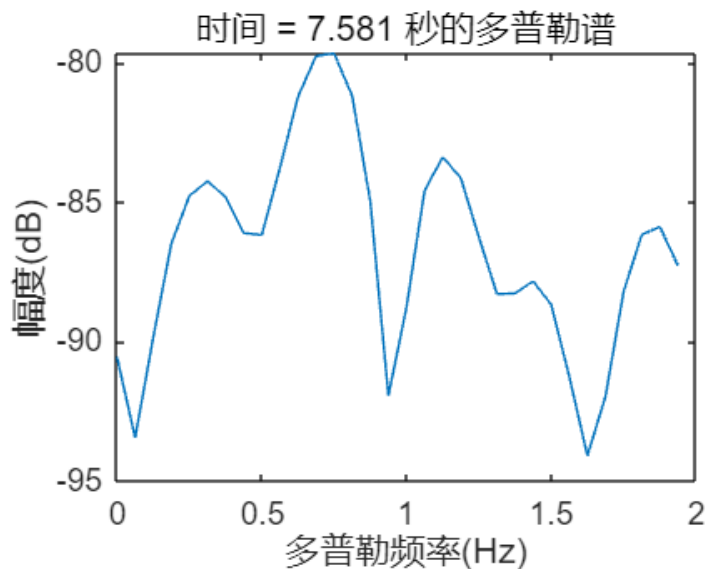
% 插值后多普勒谱随时间变化动画
for i = 1:10:length(T_fine)
    figure(100);

```

```

plot(F, 20*log10(S_fine(:,i)));
xlabel('多普勒频率(Hz)');
ylabel('幅度(dB)');
title(sprintf('时间 = %.3f 秒的多普勒谱', T_fine(i)+cur_time_all(1)));
set(gca, 'FontSize', 12);
pause(0.1);
end

```



结论

通过提高 STFT 的时间分辨率和使用插值方法细分时间轴，可以显著提高时间-多普勒图的时间分辨率。这不仅有助于更精确地分析目标的运动特性，还可以为雷达信号处理提供更丰富的信息。这些技术的应用可以扩展到更广泛的领域，如目标识别、跟踪和分类等，从而提高雷达系统的性能和可靠性。

总结：

1.问题：滤波器参数设置不当可能会截断有效信号或者保留干扰

经验：通带太窄，原始信号幅度变弱，波形失真；通带太宽，滤完仍有高频杂波、干扰峰未抑制，频谱混乱。因此必须先要必须了解目标信号的频谱范围，通带频率保守覆盖，阻带留有缓冲，所以先画频率响应，再滤信号

2.问题：时间分辨率与频率分辨率的权衡

在调整 spectrogram 参数时发现，减小窗口步长（提高时间分辨率）会导致频率分辨率下降，出现频谱模糊现象。

经验：需根据目标运动特性（快速机动/匀速）选择合适的窗口长度和重叠率，高速运动目标需要更高时间分辨率。

3.问题：插值引入的伪影

使用样条插值细分时间轴时，高频区域出现虚假峰值。

经验：插值前需对原始频谱进行平滑处理（如 Savitzky-Golay 滤波），并限制插值倍数（通常 ≤ 10 倍）。