

Project2 : Motion detection via communication signals

涂峻陵(12213010), 欧阳莹轩(12212961), 肖星辰(12212904), 欧阳安男(12211831)

Contents

Introduction.....	1
Results and Analysis.....	3
Task 1.....	3
TASK1.....	7
Description.....	7
Task 2.....	7
task2.....	9
Description.....	9
Task 3.....	9
Description.....	10
Enhancement.....	11
Task 2.....	11
Task 2.....	12
Task 3.....	14
describe.....	15
Functions.....	15
Expeience.....	17

Introduction

1、掌握时域处理方法：

(1) 数字下变频 Digital Down Convert

- 定义采样时间 t 为从 $1/f_s$ 到 0.5 的时间序列，其中 f_s 为采样频率。
- 定义下变频的频率 f_{ddc}
- 对参考信号 ref 和监视信号 sur 进行数字下变频，即在时域上通过点乘 $\exp(-1i * 2\pi * f_{ddc} * t)$ 这一复指数实现频率的偏移。

(2) 数字滤波器设计及使用；

- . A. 可以使用`butter`函数来设计数字Butterworth滤波器：

a. `[b, a] = butter(n, Wn)`：可以设计一个阶次为 n 的低通数字Butterworth滤波器，其归一化截止频率为 Wn 。其中， Wn 的取值范围为 $0.0 < Wn < 1.0$ ，其中 1.0 对应采样率的一半。返回滤波器系数分别存储在长度为 $n+1$ 的向量 b 和 a 中。

b. `[b, a] = butter(n, Wn, 'ftype')`: 可以用于设计高通、低通或带阻数字Butterworth滤波器。通过字符串参数'ftype'指定滤波器类型, 可以选择'high' (高通)、'low' (低通) 或'stop' (带阻)。同样, 返回滤波器系数分别存储在长度为 $n+1$ 的向量**b**和**a**中。

B. 可以使用**freqz** 函数计算数字滤波器的频率响应:

例如 `[H, w] = freqz(b, a, 512);`

其中**freqz** 函数的输入参数包括由**butter** 函数设计的滤波器的系数 **b** 和 **a**。**512** 指定了计算频率响应时的数据点数量, 可以根据需要进行调整。计算完成后, **freqz** 返回频率响应 **H** 和对应的频率向量 **w**。

(3) 模糊函数 Ambiguity Function 计算 (互相关)

模糊函数提供了衡量信号分量或目标回波之间模糊或不确定性的度量, 特别是在雷达或通信系统中使用于分析信号的时频表示。具体而言, 它描述了信号能量在时间和频率领域中的分布情况。在雷达系统中 用于研究与不同目标相关的范围和多普勒频移的模糊性。

模糊函数的设计步骤:

A. 初始化

a. 定义了一个MATLAB函数接受一系列参数 (包含 τ 、 fd 、 T 、 N 、 seq_sur 、 seq_ref) 并返回模糊函数矩阵。

b. 初始化待返回的矩阵用于存储计算结果。定义矩阵大小行数为 τ 的长度, 列数为 fd 的长度。

c. 通过总时间长度 T 和采样点数 N 计算采样周期 T_s 和采样频率 f_s 。

B. 循环

a. 通过一个for循环遍历 τ 的所有取值。

b. 在每个 τ 值下, 再次进行for循环, 遍历 fd 的所有取值。

c. 判断 τ 是否为零。

若 τ 为零, 说明不需要进行时域平移。将参考信号 seq_ref 取共轭赋值给 ref_shift , 而观测信号 seq_sur 直接赋值给 sur_shift 。

若 τ 不为零, 说明需要进行时域平移。通过将 τ 除以采样周期 T_s 并四舍五入得到平移的点数, 在参考信号和观测信号前后分别添加需要的零点, 实现平移操作。

C. 计算与返回

a. 使用了傅里叶变换计算公式下模糊函数的值。

b. 对模糊函数矩阵乘以采样周期 T_s , 以获得最终的模糊函数值。

c. 返回包含了不同时延 (τ) 和多普勒频移 (fd) 下的模糊函数值的矩阵。

2、掌握频率处理方法:

(1) 分析信号频谱;

- a.定义频谱的范围。
- b.对原始信号和经过数字下变频和低通滤波的信号进行傅里叶变换，得到相应的频谱。
- c.通过绘制频谱图，可以直观地分析信号在频率域上的特征。

(2) 低通滤波设计及验证；

A.确定滤波器参数

- a.定义低通滤波器截止频率。
- b.在所需要的频率范围内设计滤波器
- c.设置频段数量为N

B.使用滤波器设计函数

通过函数**`[b, a] = butter(N, frequency_cutoff/(fs/2), 'low');`**设置低通滤波器

C.滤波器函数验证

创建测试信号，应用滤波器并分析滤波前后的信号，确保滤波器能够正确地滤除或保留所需的频率成分，进行验证。

D.应用滤波器

在对于频段*i*上对于信号*s*使用 **filter** 函数：**`y = filter(b, a, s);`**

3、基于通信信号的运动检测技术在生活中的应用

主要原理是针对通信信号进行处理和分析，提取运动特征，并实现相应的应用。

1.智能监控系统：通过智能监控摄像头进行基于通信信号的运动检测技术，分析视频中的运动信号来检测人物行为以及可能的异常情况。

2.健康监测：通过分析基站信号或其他通信信号，可以实现基于无线通信的健康监测系统。例如，通过监测信号的运动特征，可以检测到人体的呼吸和心跳。

3.智能交通系统：通过分析车辆或行人产生的通信信号，可以实现智能交通系统，监测交通流量并进行智能信号灯控制。

Results and Analysis

Task 1

- For the reference signal and the surveillance signal, draw their time domain waveforms and their frequency domain waveforms.

Initialize and load audio to *ref* and *sur*.

```
clear;clc;
load('data_1.mat');
t = 1/f_s : 1/f_s : 0.5;
t_ms = t*1000;
ref = seq_ref;
sur = seq_sur;
```

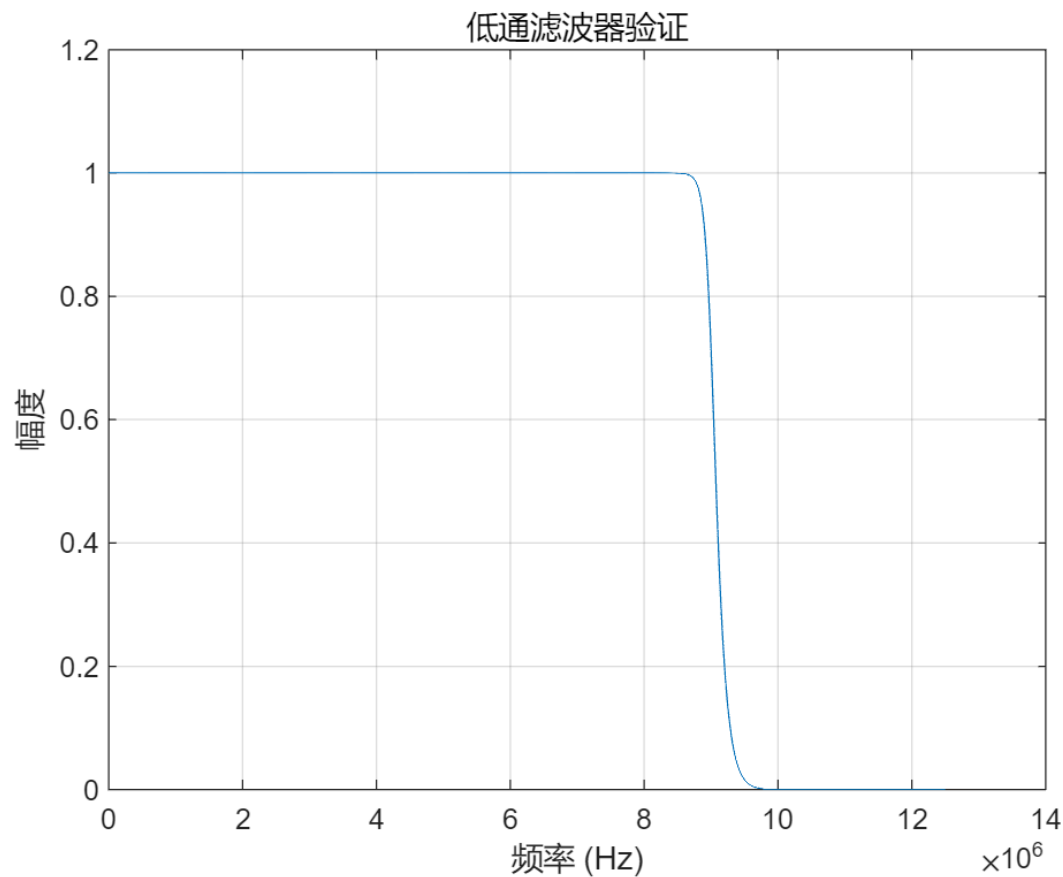
Do Digital Down Convert (DDC).

```
f_ddc = -3e6;
ref_ddc = ref .* exp(-1i * 2*pi * f_ddc * t);
sur_ddc = sur .* exp(-1i * 2*pi * f_ddc * t);
```

Do Low Pass Filter (LPF).

```
f_cutoff = 9e6;
[b,a] = butter(24, f_cutoff/(f_s/2));
[h, w] = freqz(b, a, 8000);

% 绘制频域图
plot(w*f_s/(2*pi), abs(h))
title('低通滤波器验证')
xlabel('频率 (Hz)')
ylabel('幅度')
grid on
```



```
ref_lpf = filter(b, a, ref_ddc);
sur_lpf = filter(b, a, sur_ddc);
```

Calculate spectrum.

```
freq_mhz = 2110-2123+2e-6 : 2e-6 : 2135-2123;
sur_freq = abs(fftshift(fft(sur)));
sur_ddc_freq = abs(fftshift(fft(sur_ddc)));
sur_lpf_freq = abs(fftshift(fft(sur_lpf)));
ref_freq = abs(fftshift(fft(ref)));
ref_ddc_freq = abs(fftshift(fft(ref_ddc)));
ref_lpf_freq = abs(fftshift(fft(ref_lpf)));
```

Convert to dB.

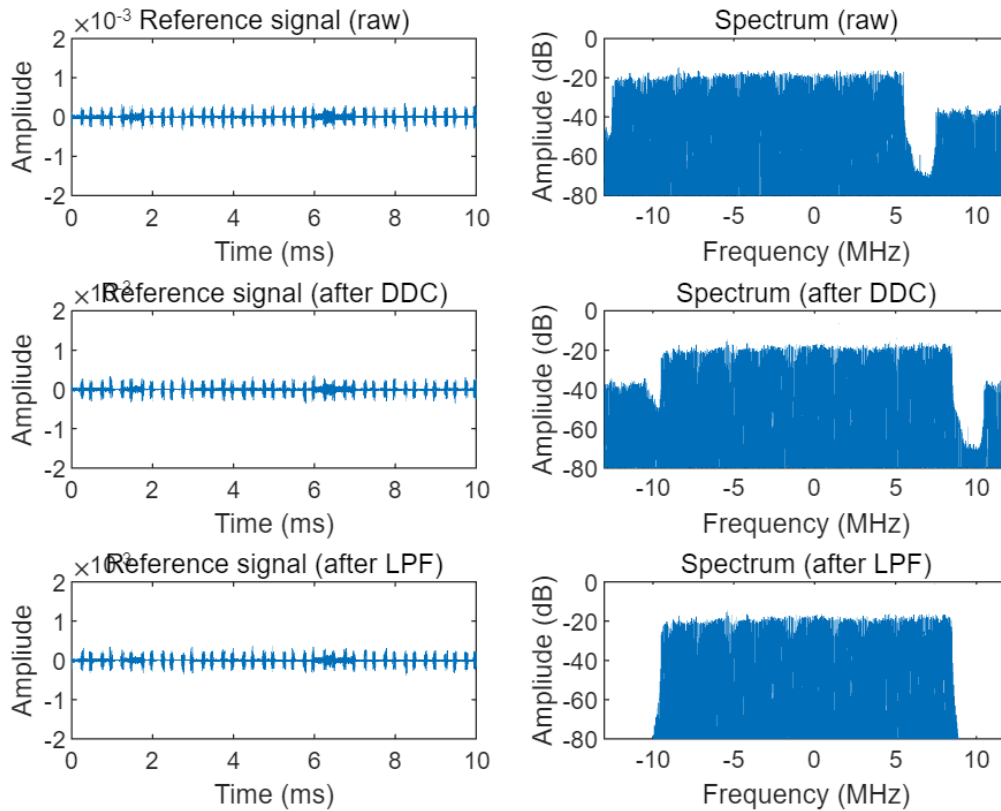
```
ref_freq = 20 * log10(ref_freq / max(sur_freq));
ref_ddc_freq = 20 * log10(ref_ddc_freq / max(sur_ddc_freq));
ref_lpf_freq = 20 * log10(ref_lpf_freq / max(sur_lpf_freq));
sur_freq = 20 * log10(sur_freq / max(sur_freq));
sur_ddc_freq = 20 * log10(sur_ddc_freq / max(sur_ddc_freq));
sur_lpf_freq = 20 * log10(sur_lpf_freq / max(sur_lpf_freq));
```

Plot.

```

figure;
task1plot(1, 'Reference signal (raw)', t_ms, real(ref), true);
task1plot(3, 'Reference signal (after DDC)', t_ms, real(ref_ddc), true);
task1plot(5, 'Reference signal (after LPF)', t_ms, real(ref_lpf), true);
task1plot(2, 'Spectrum (raw)', freq_mhz, ref_freq, false);
task1plot(4, 'Spectrum (after DDC)', freq_mhz, ref_ddc_freq, false);
task1plot(6, 'Spectrum (after LPF)', freq_mhz, ref_lpf_freq, false);

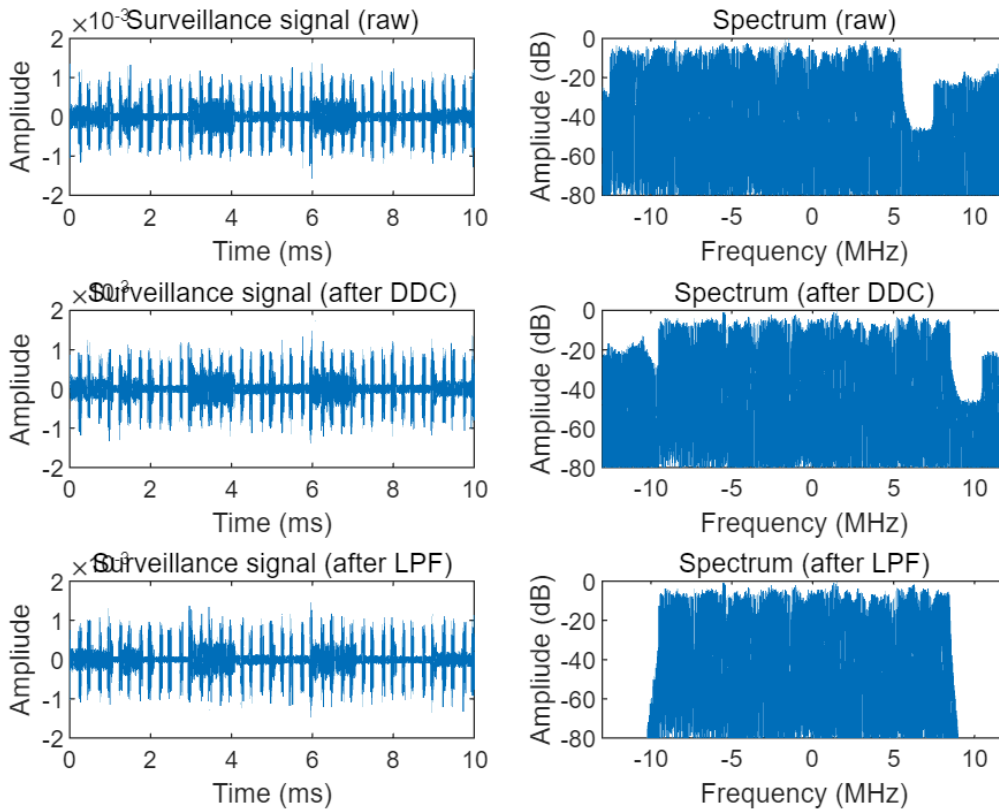
```



```

figure;
task1plot(1, 'Surveillance signal (raw)', t_ms, real(sur), true);
task1plot(3, 'Surveillance signal (after DDC)', t_ms, real(sur_ddc), true);
task1plot(5, 'Surveillance signal (after LPF)', t_ms, real(sur_lpf), true);
task1plot(2, 'Spectrum (raw)', freq_mhz, sur_freq, false);
task1plot(4, 'Spectrum (after DDC)', freq_mhz, sur_ddc_freq, false);
task1plot(6, 'Spectrum (after LPF)', freq_mhz, sur_lpf_freq, false);

```



TASK1

这道题要求对雷达接收的已调制信号进行处理，具体步骤为

DCC:将信号频谱向右频移3Mhz，使有用的带宽范围关于远点对称

通过低通滤波器: 通过一个截止频率为9Mhz的低通滤波器，滤去杂波信号

Description

观察spectrum (raw) 与spectrum (DCC) 可以得出: 经过DCC后信号的频域图整体向右移动了3Mhz

观察spectrum (after LPF) 可以得出: 频率绝对值大于9Mhz的杂波信号已经被滤去

Task 2

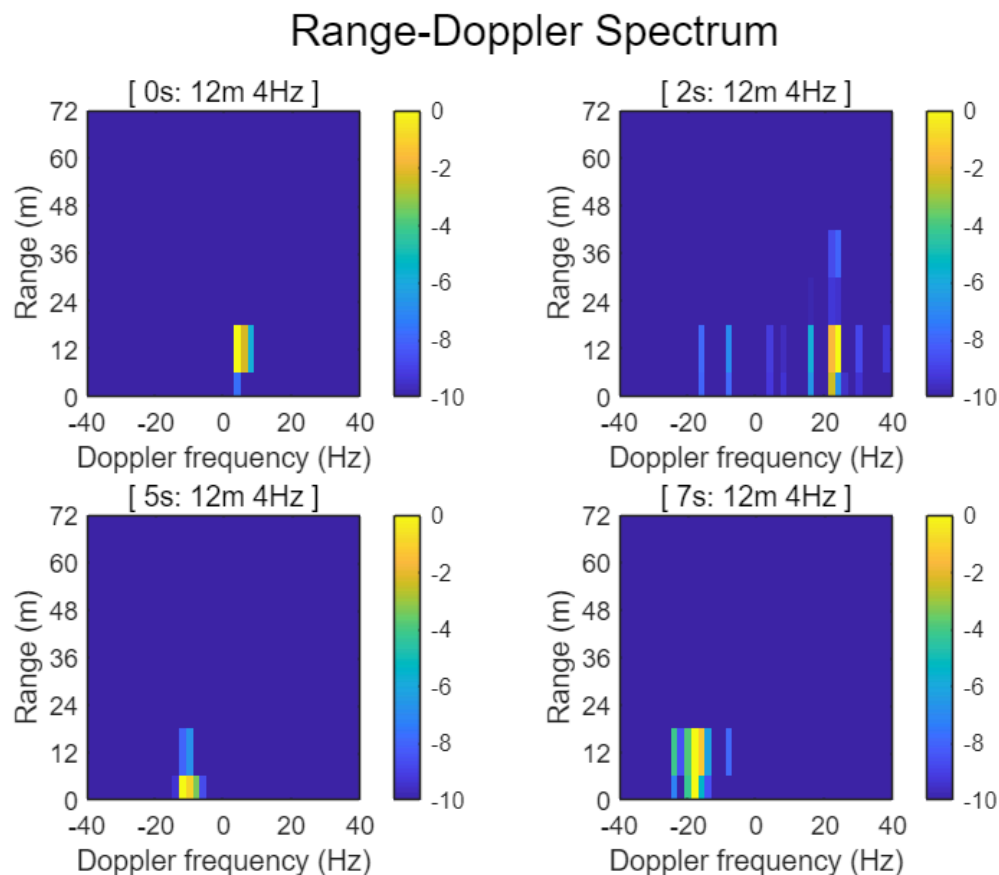
- Plot the Range Doppler spectrum of the signal located at 0-0.5s, 2-2.5s, 5-5.5s and 7-7.5s.

Initialize the light speed c and the range of τ and f_D .

```
clear;clc;
light_speed = 3e8;
tau = 0 : 12/light_speed : 72/light_speed;
fd = -40:2:40;
```

Calculate the cor for every τ and f_D in range, and plot them in the images of Range-Doppler Spectrum.

```
figure;
sgtitle('Range-Doppler Spectrum');
load('proj2/data_1.mat');
sur = ddc_lpf(seq_sur, -3e6, 9e6, f_s, 0.5);
ref = ddc_lpf(seq_ref, -3e6, 9e6, f_s, 0.5);
cor1 = Cor(tau, fd, 0.5, 125e5, sur, ref);
Draw_task2(tau, fd, cor1, 1, 0.0);
load('proj2/data_5.mat');
sur = ddc_lpf(seq_sur, -3e6, 9e6, f_s, 0.5);
ref = ddc_lpf(seq_ref, -3e6, 9e6, f_s, 0.5);
cor2 = Cor(tau, fd, 0.5, 125e5, sur, ref);
Draw_task2(tau, fd, cor2, 2, 2.0);
load('proj2/data_11.mat');
sur = ddc_lpf(seq_sur, -3e6, 9e6, f_s, 0.5);
ref = ddc_lpf(seq_ref, -3e6, 9e6, f_s, 0.5);
cor3 = Cor(tau, fd, 0.5, 125e5, sur, ref);
Draw_task2(tau, fd, cor3, 3, 5.0);
load('proj2/data_15.mat');
sur = ddc_lpf(seq_sur, -3e6, 9e6, f_s, 0.5);
ref = ddc_lpf(seq_ref, -3e6, 9e6, f_s, 0.5);
cor4 = Cor(tau, fd, 0.5, 125e5, sur, ref);
Draw_task2(tau, fd, cor4, 4, 7.0);
```



task2

这道题要求通过模糊函数求出接收信号的时间差 τ 和多普勒频移 f_D

Description

将信号经过tsak1的要求进行DCC与滤波后，使用模糊函数cor处理

在模糊函数中cor中将 f_D 和 τ 遍历，并将结果储存在一个矩阵中

然后求矩阵中最大的数据点，该点对应的的时间差以及多普勒频移即被认为是此时真实的时间差和多普勒频移

根据时间差求出此时该学生的位置，观察结果图像可知，在时刻0s, 2s, 5s, 7s, cor函数中的矩阵均有显著且唯一极值，可知用模糊函数处理信号的方式是合理的，且求出来的位置和多普勒频移是准确的，据此可以求出各个时刻的位置和速度，速度以朝雷达奔跑为正方向：

0s时距离雷达12m左右，速度约为0m/s；

2s时距离雷达12m左右，速度约为3.15m/s；

5s时距离雷达0m左右，速度约为-1.65m/s；

7s时距离雷达12m左右，速度约为-3m/s；

Task 3

- Plot the Time Doppler spectrum with coherent integration time (CIT) equal to 0.5s. Time range is 0-9.5s.

Initialize the light speed c , time range $time$ and the range of τ and f_D .

```
clear;clc;
light_speed = 3e8;
tau = 0 : 12/light_speed : 72/light_speed;
fd = -40:2:40;
time = 0:0.5:9.5;
```

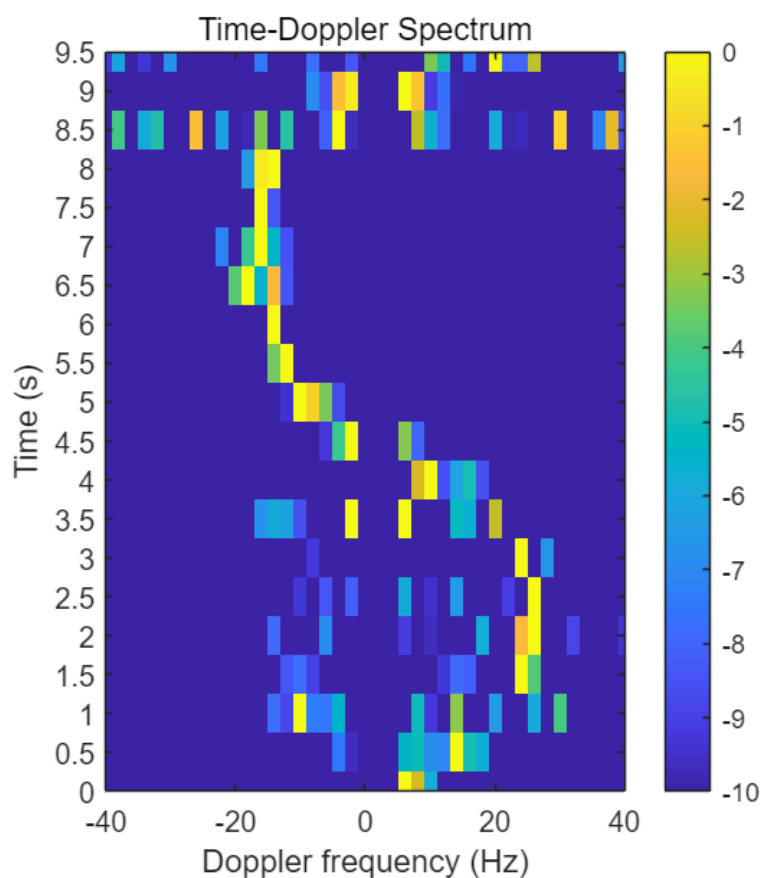
Calculate the cor for every f_D in range and for τ which makes cor gets the maximum.

```
g = zeros(length(time),length(fd));
for data_index = 1:20
    load("proj2/data_"+data_index+".mat");
    sur = ddc_lpf(seq_sur, -3e6, 9e6, f_s, 0.5);
    ref = ddc_lpf(seq_ref, -3e6, 9e6, f_s, 0.5);

    cor = Cor_fft(tau, fd, 0.5, 125e5, sur, ref);
    corMax = max(max(cor));
    [tauMax_index, ~] = find(corMax==cor);
    cor_relative = 10.*log(cor./max(cor(tauMax_index,:)));
    g(data_index,:) = cor_relative(tauMax_index,:);
end
```

Plot them in the images of Time-Doppler Spectrum. Time range holds for 0.0s-9.5s.

```
figure;
img = imagesc(fd, time, real(g));
clim([-10 0]);
colorbar
set(gca,'Ydir','normal');
title("Time-Doppler Spectrum");
xlabel("Doppler frequency (Hz)");
ylabel("Time (s)");
axis([-40 40 0.0 9.5]);
xticks(-40:20:40);
yticks(0.0:0.5:9.5);
img.Parent.DataAspectRatio = [12 1 1];
```



Description

文件中以0.5s为单位分成了20个文件，据此采用循环绘制出0-9.5s范围内的多普勒频移图像，基带信号为2000Mhz左右，计算得波长大约为0.15m，速度约等于 $0.15 \cdot f_D$ ，观察图像可知，测试者先朝着雷达奔跑，速度达到约4.5m/s的峰值后逐渐减速，减速至0m/s的时候开始朝反方向奔跑，达到峰值速度约-3m/s时停下

Enhancement

Task 2

- In this part, we calculate the *cor* by the help of *DFT* to enhance.

Initialize the light speed c and the range of τ and f_D .

```
clear;clc;
light_speed = 3e8;
tau = 0 : 12/light_speed : 72/light_speed;
fd = -40:2:40;
```

Calculate the *cor* for every τ and f_D in range, and plot them in the images of Range-Doppler Spectrum.

```
figure;
sgtitle('Range-Doppler Spectrum');
load('proj2/data_1.mat');
tic;
sur = ddc_lpf(seq_sur, -3e6, 9e6, f_s, 0.5);
ref = ddc_lpf(seq_ref, -3e6, 9e6, f_s, 0.5);
cor1 = Cor(tau, fd, 0.5, 125e5, sur, ref);
toc;
```

历时 86.346327 秒。

```
Draw_task2(tau, fd, cor1, 1, 0.0);
tic;
cor1 = Cor_fft(tau, fd, 0.5, 125e5, sur, ref);
toc;
```

历时 7.875763 秒。

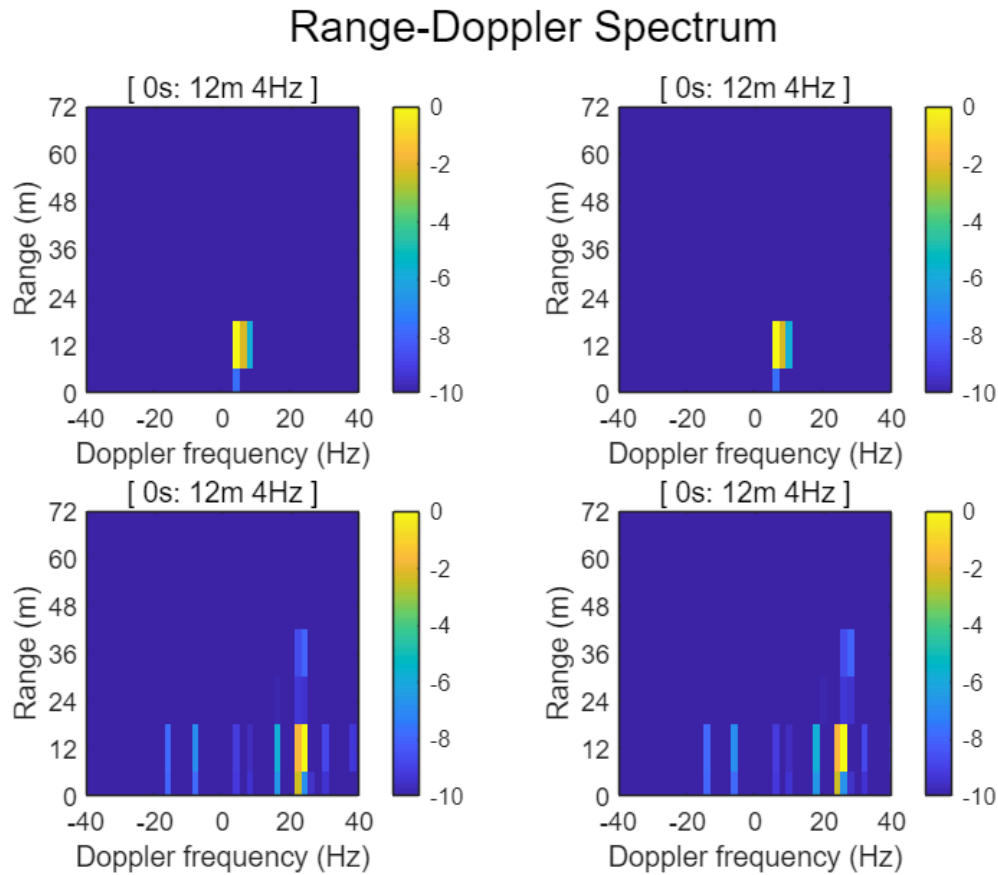
```
Draw_task2(tau, fd, cor1, 2, 0.0);
load('proj2/data_5.mat');
tic;
sur = ddc_lpf(seq_sur, -3e6, 9e6, f_s, 0.5);
ref = ddc_lpf(seq_ref, -3e6, 9e6, f_s, 0.5);
cor2 = Cor(tau, fd, 0.5, 125e5, sur, ref);
toc;
```

历时 86.533992 秒。

```
Draw_task2(tau, fd, cor2, 3, 0.0);
tic;
cor2 = Cor_fft(tau, fd, 0.5, 125e5, sur, ref);
toc;
```

历时 8.487225 秒。

```
Draw_task2(tau, fd, cor2, 4, 0.0);
```



By using *fft*, we can save as much as 10 times of time to calculate the *cor*, and they have the same effect.

describe

在*cor*函数中使用*dft*求和，尝试用*fft*优化计算，经过画图检验后发现*fft*绘图的结果和*dft*绘图的结果一致，并且速度提升了10倍左右，可以有效的减少计算时间

Task 2

- In this part, we try to abandon the filter of Digital Down Convert (DDC) and Low-Pass Filter (LPF), to order to discuss the effect of them.

Initialize the light speed c and the range of τ and f_D .

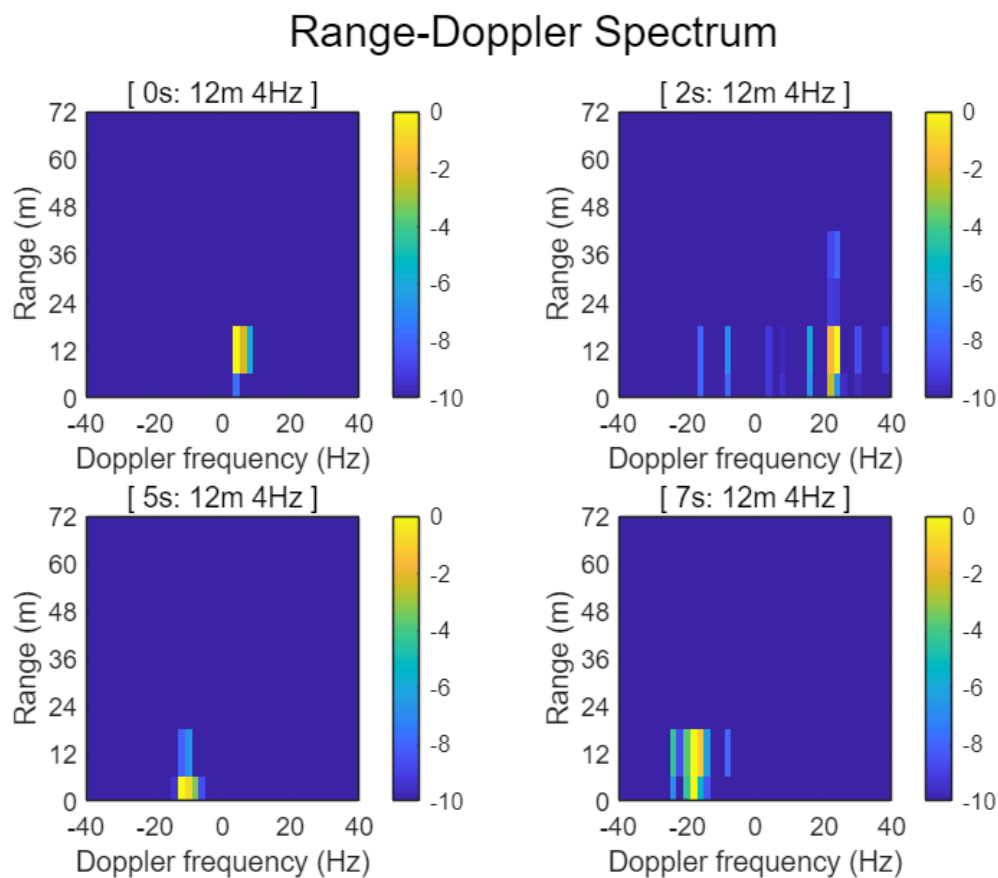
```
clear;clc;
light_speed = 3e8;
tau = 0 : 12/light_speed : 72/light_speed;
fd = -40:2:40;
```

Calculate the *cor* for every τ and f_D in range, and plot them in the images of Range-Doppler Spectrum.

```

figure;
sgtitle('Range-Doppler Spectrum');
load('proj2/data_1.mat');
cor1 = Cor(tau, fd, 0.5, 125e5, seq_sur, seq_ref);
Draw_task2(tau, fd, cor1, 1, 0.0);
load('proj2/data_5.mat');
cor2 = Cor(tau, fd, 0.5, 125e5, seq_sur, seq_ref);
Draw_task2(tau, fd, cor2, 2, 2.0);
load('proj2/data_11.mat');
cor3 = Cor(tau, fd, 0.5, 125e5, seq_sur, seq_ref);
Draw_task2(tau, fd, cor3, 3, 5.0);
load('proj2/data_15.mat');
cor4 = Cor(tau, fd, 0.5, 125e5, seq_sur, seq_ref);
Draw_task2(tau, fd, cor4, 4, 7.0);

```



describe:

观察task1的结果，可以发现信噪比在20db左右，噪音的强度较低，观察模糊函数的公式，一个是时间差，一个是只与多普勒频移有关的旋转矢量

我们推测在题目的条件下不进行DCC和滤波不会影响结果

最后，将直接处理的结果图像和task2的图像比较，我们可以得出结论，不进行DCC和滤波，对模糊函数的处理结果几乎没有影响

Task 3

- In this part, we try to set coherent integration time (CIT) as 0.1s.

Initialize the light speed c , time range $time$ and the range of τ and f_D .

```
clear;clc;
light_speed = 3e8;
tau = 0 : 12/light_speed : 72/light_speed;
fd = -40:2:40;
time = 0:0.5:9.5;
```

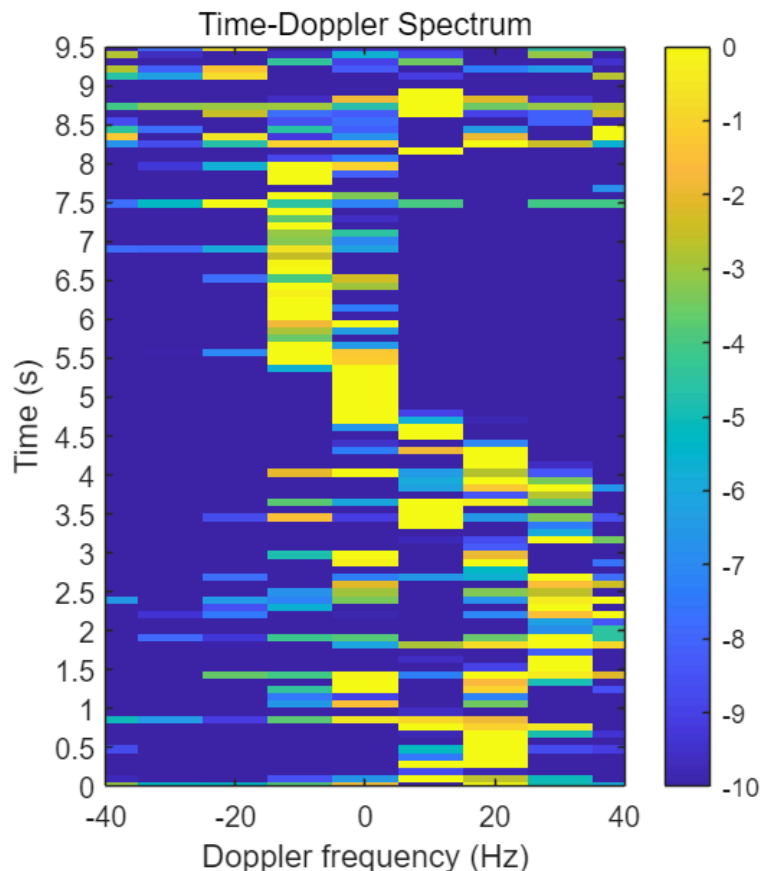
Calculate the cor for every f_D in range and for τ which makes cor gets the maximum.

```
g = zeros(length(time),length(fd));
for data_index = 1:20
    load("proj2/data_"+data_index+".mat");
    for div_time = 1:5
        cor = Cor_fft(tau, fd, 0.5, 125e5, seq_sur(1+length(seq_sur)/5*(div_time-1):length(seq_sur)));
        tauMax_index = 1;
        corMax = cor(1);
        for index_tau = 1:length(tau)
            for index_fd = 1:length(fd)
                if corMax < cor(index_tau, index_fd)
                    corMax = cor(index_tau, index_fd);
                    tauMax_index = index_tau;
                end
            end
        end
        corMax = max(max(cor));
        cor_relative = 10.*log(cor./corMax);
        g((data_index-1)*5+div_time,:) = cor_relative(tauMax_index,:);
    end
end
```

Plot them in the images of Time-Doppler Spectrum. Time range holds for 0.0s-9.5s.

```
figure;
img = imagesc(fd, time, real(g));
clim([-10 0]);
colorbar
set(gca,'Ydir','normal');
title("Time-Doppler Spectrum");
xlabel("Doppler frequency (Hz)");
ylabel("Time (s)");
axis([-40 40 0.0 9.5]);
xticks(-40:20:40);
yticks(0.0:0.5:9.5);
```

```
img.Parent.DataAspectRatio = [12 1 1];
```



From the image, we can tell that, since the CIT becomes 0.1s, the f_D affects less, that is, with a more accurate CIT, we can get a less accurate f_D .

describe

观察模糊函数的公式可知，因为采样频率是固定的，所以当时间间隔变短即公

$$Cor(\tau, f_D, t_0) = \sum_{n=0}^{N-1} y_{sur}(nT_s) y_{ref}^*(nT_s - \tau) e^{-j2\pi f_D nT_s} \quad N-1$$

的数量会减小，而数据点数量的减小会使 f_D 的分辨率变差，而时域上有更多的数据点会使时域的分辨率变高。
式中的 N 的值减小时，多普勒频移对应的复指数项的数据点

Functions

The function to plot in task1.

```
function task1plot(plot_idx, title_name, x, y, is_t_domain)
subplot(3, 2, plot_idx);
plot(x, y);
title(title_name);
if is_t_domain
    xlabel('Time (ms)');
    ylabel('Amplitude');
```

```

    axis([0 10 -2e-3 2e-3]);
    xticks(0:2:10);
    yticks((-2:2)*1e-3);
else
    xlabel('Frequency (MHz)');
    ylabel('Amplitude (dB)');
    axis([-13 12 -80 0]);
    yticks(-80:20:0);
end
end

```

Function to get a signal through Digital Down Convert (DDC) and Low-Pass Filter (LPF).

```

function res = ddc_lpf(sig, f_ddc, f_cutoff, fs, T)
t = 1/fs : 1/fs : T;
[b, a] = butter(24, f_cutoff/(fs/2));
sig_ddc = sig .* exp(-1i * 2*pi * f_ddc * t);
res = filter(b, a, sig_ddc);
end

```

Function to $Cor(\tau, f_D, t_0)$

```

function cor = Cor(tau, fd, T, N, seq_sur, seq_ref)
cor = zeros(length(tau), length(fd));
Ts = T/N;
fs = 1/Ts;
for index_tau = 1:length(tau)
    for index_fd = 1:length(fd)
        if(tau(index_tau) == 0)
            ref_shift = conj(seq_ref);
            sur_shift = seq_sur;
        else
            shift = round(tau(index_tau)/Ts);
            ref_shift = conj([ zeros(1, shift), seq_ref ]);
            sur_shift = [ seq_sur, zeros(1, shift) ];
        end
        cor(index_tau, index_fd) = sum( sur_shift .* ref_shift .* exp(-1i*2*pi*(fd(index_fd)/fs * tau(index_tau))) );
    end
end

cor = cor.*Ts;
end

```

Function enhanced to $Cor(\tau, f_D, t_0)$ by using *fft*.

```

function cor = Cor_fft(tau, fd, T, N, seq_sur, seq_ref)
cor = zeros(length(tau), length(fd));
Ts = T/N;
fs = 1/Ts;
for index_tau = 1:length(tau)
    offset = round(tau(index_tau)/Ts);

```



```

y = [seq_sur, zeros(1,offset)] .* [zeros(1,offset), conj(seq_ref)];

fft_y = fftshift(fft(y));
indices = round(length(seq_sur).*fd/fs + length(seq_sur)/2);
cor(index_tau,:) = fft_y(1,indices);
end
end

```

Function draw Range-Doppler Spectrum for task2.

```

function Draw_task2(tau, fd, cor, pic_index, time_start)
corMax = max(max(cor));
cor_relative = 10.*log(cor./corMax);

light_speed = 3e8;
subplot(2,2,pic_index)
imagesc(fd, tau*light_speed, real(cor_relative));
clim([-10 0]);
colorbar
set(gca, 'Ydir', 'normal');
title("[ "+time_start+"s: 12m 4Hz ]");
xlabel("Doppler frequency (Hz)");
ylabel("Range (m)");
axis([-40 40 0 72]);
xticks(-40:20:40);
yticks(0:12:72);
end

```

Expeience

1.在本次project中，我们学习根据无源雷达的基本原理，借助通信系统的时域处理与频率分析工具，通过MATLAB对数据进行处理分析，获得有效信息，成功地实现了对物体位置和运动信息的分析。这为我们在雷达信号处理领域的知识储备提供了坚实的基础，同时也拓展了我们在实际应用中解决问题的能力。

2.在project的拓展任务中，我们通过尝试和实验得出结论：①fft绘图的结果和dft绘图的结果一致，并且速度提升了10倍左右，可以有效的在保证结果正确的前提下减少很大程度上的计算时间。②不进行DCC和滤波，对模糊函数的处理结果影响较小。③为采样频率是固定的，所以当时间间隔变短时，多普勒频移对应的复指数项的数据点的数量会减小，而数据点数量的减小会使 f_D 的分辨率变差，而时域上有更多的数据点会使时域的分辨率变高。

3. 在项目进展中，我们采用了小组分工合作的模式，每位成员根据个人的专业领域和技能贡献力量。这种协同合作不仅加强了小组成员之间的团队凝聚力，还提升了个体在协调沟通方面的能力。通过共同努力，我们成功

地完成了项目，并从中汲取了团队协作的宝贵经验，为未来更复杂的团队任务奠定了坚实的基础。这次合作不仅是知识的传递与分享，更是团队协作的实践与提升，为我们未来的职业生涯奠定了坚实的基础。

4.贡献比：涂峻綾(12213010) 25% 欧阳莹轩(12212961) 25% 肖星辰(12212904) 25% 欧阳安男(12211831) 25%

.