<Project 2> Motion Detection via Communication Signals

<Author>

Lifer XD

<Introduction>

Passive Radar (PCL)

The expression “passive radar systems” indicates a class of bistatic radar systems that do not transmit a dedicated electromagnetic signal, but instead exploit electromagnetic signals emitted by other sources for other purposes. Since the illuminating signal is not a priori known to the passive radar receiver, it has to measure both the signal to be exploited as well as the signal reflected off the target. Further, the radar has to preprocess the two signals, cross-correlate them, detect the targets, and finally track the targets.

Multipath Propagation

The signal from the base station is transmitted in two ways and received by passive radar. One is a reference channel, transmitted directly from the base station to the receiving radar. The other is a signal sent from the base station to the target object, and from the target object to the receiving radar. The signals of these two paths are respectively referred to as reference signals and research signals. The expression of the two signals is a reference channel, directly transmitted from the base station to the receiving radar. The other is a signal sent from the base station to the target object, and from the target object to the receiving radar. The signals of these two paths are respectively referred to as reference signals and survey signals.

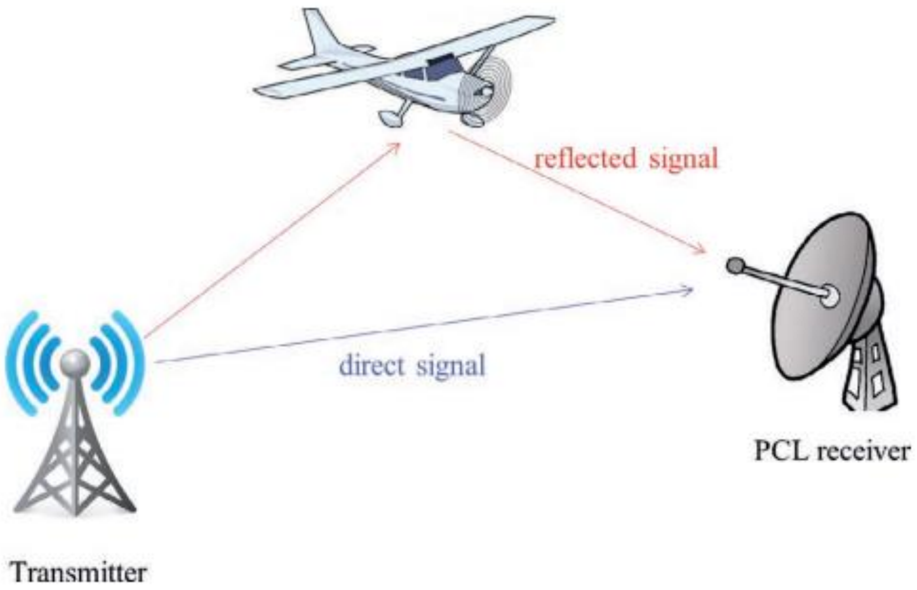

Figure 1 Multipath Propagation

Denote the signal as $x(t)$, we can define that:

$$y_{ref}(t) = \alpha x(t - \tau_r)$$

$$y_{Sur}(t) = \beta x(t - \tau_s)e^{j2\pi f_d t}$$

These two signals have time differences and frequency differences.

In time domine:

$$\tau_s - \tau_r = \Delta d \,/\, C$$

In frequency domine: we use doppler shift.

Doppler Shift

When the source and observer of a signal are moving relatively, the frequency will change. Increase when approaching (which is called a “blue shift”), decreasing when moving apart (which is called a “red shift”).

The wave source will send out a wave after vibration, and the receiver will receive a different frequency due to relative movement.

The Doppler frequency change follows the formula:

$$f' = \left(\frac{v \pm v_0}{v \mp v_s}\right) f$$

Project Objective

The pictures of the scene are as follows:

Figure 2 Signal Collection Site

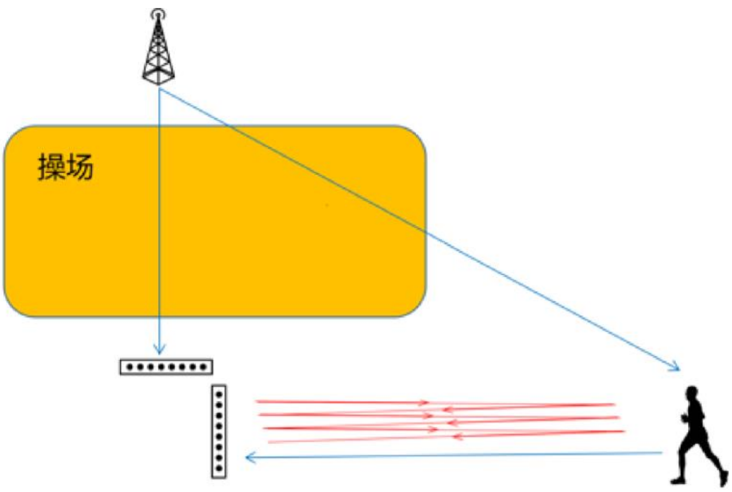

Figure 3 Signal Acquisition Demonstration Diagram

As is shown in the picture, the mountain is the base station of this time, the laboratory has built a receiving radar, and people are the objects to be monitored. We need to understand human movement through the analysis of two signals from the receiving radar.

<Methodology>

What Is $\Delta\tau$ and f_D

We defined $y_{ref}(t)$ and $y_{sur}(t)$. And we need to figure out $\Delta\tau$ and f_D.

The time shift between two signals is

$$\Delta\tau = |\tau_r - \tau_s|$$

and we can get ΔL by multiplying Δt with the light speed c.

$$\Delta L = \Delta\tau \cdot c$$

f_D is the doppler frequency shift.

We can calculate v while using formula:

$$v = \frac{\lambda \cdot f_D}{2cos\frac{\beta}{2}}$$

for the β, we can measure $L = |OA|$ which is 247 meters, also $x(t) = |OB|$, then we have $|AB| = \sqrt{L^2 + x^2(t)}$.

The distance different is $|AB| + |BO| - |AO|$.

So, we can know:

$$\sqrt{L^2 + x^2(t)} + x(t) - L = \Delta L = \Delta\tau \cdot c$$

$$\beta = tan^{-1}\frac{L}{x(t)}$$

Then we can use the formulas above to figure out $v(t)$ by using $\Delta\tau$ and f_D.

How to Find $\Delta\tau$ and f_D

We use the Cross-correlation Function (CCF), which is related to the self-correlation Function we used in echo cancellation.

We define the ambiguity function as:

$$Cor(\tau, f_D) = \int_{t}^{t+T} y_{sur}(t) y_{ref}^{*}(t-\tau) e^{-j2\pi f_D t} dt$$

For a DT-signal system, we can define CCF as below:

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

When $\Delta\tau$ and f_D close to the real value, $Cor(\tau, f_D)$ gets bigger.

So, we want to get $(\Delta\tau, f) = argmaxCor(c, d)$.

< Results & Analysis>

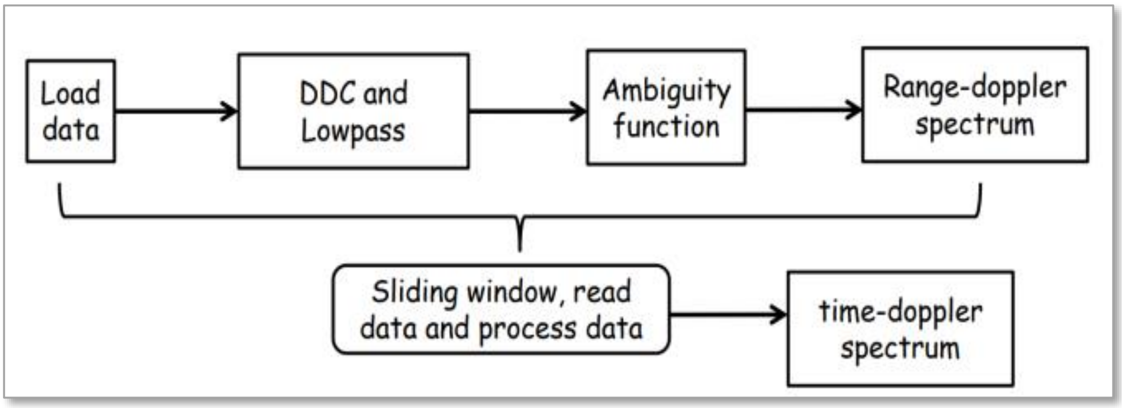

Figure 4 The Process of Lab

Task 1. Plotting the Spectrums

Step 1. Load Data

The radar receives the baseband signals distributed majorly in two frequency bands: 2110 ~ 2130 MHz (20 MHz bandwidth) and 2130 ~ 2135 MHz (5 MHz bandwidth). In this step, we need to read data from the package provided as .mat files, from data_1.mat to data_20.mat, using the function load.

```
% Load
load data_1.mat
```

Figure 5 Load Data

And for the convenience of drawing, we extract the first 0.01s as an example.

Step 2. Digital Down Convert

Digital Down Convert (DDC) can eliminate imbalance-related distortion created by an analog IF mixer and it avoids phase distortion from analog filters. After the DDC, the sample rate is significantly reduced, and we can have a more efficient implementation of the DSP routines that further process the data.

Which we need is the information in the stronger band, i.e., in this case is the 2110 ~ 2130 MHz band. So, in this step we use DDC to move the center of the 2110 ~ 2130 MHz band to the origin point in the frequency domain. This is the preparation for low pass filtering later.

```
t = linspace(0,0.01,part);
ref_shift = ref .* exp(1i*2*pi*f0.*t);
```

```
t = linspace(0,0.01,part);
sur_shift = sur .* exp(1i*2*pi*f0.*t);
```

Figure 6 Digital Down Convert

Step 3. Low Pass Filtering

Next, we use LPF to extract the information in the 2110 ~ 2130 MHz band. Then, the base station signals of 2130 ~ 2135MHz (5M bandwidth) shall be filtered, while signals of 2110 ~ 2130 MHz (20M bandwidth) are going to be retained.

```
% LPF 低通的截止频率为0.9*10^7
[b,a] = butter(10,0.88*10^7*2/f_s,'low');
ref_shift_butter = filter(b,a,ref_shift);
sur_shift_butter = filter(b,a,sur_shift);
```

Figure 7 Low Pass Filtering

Here is the result:

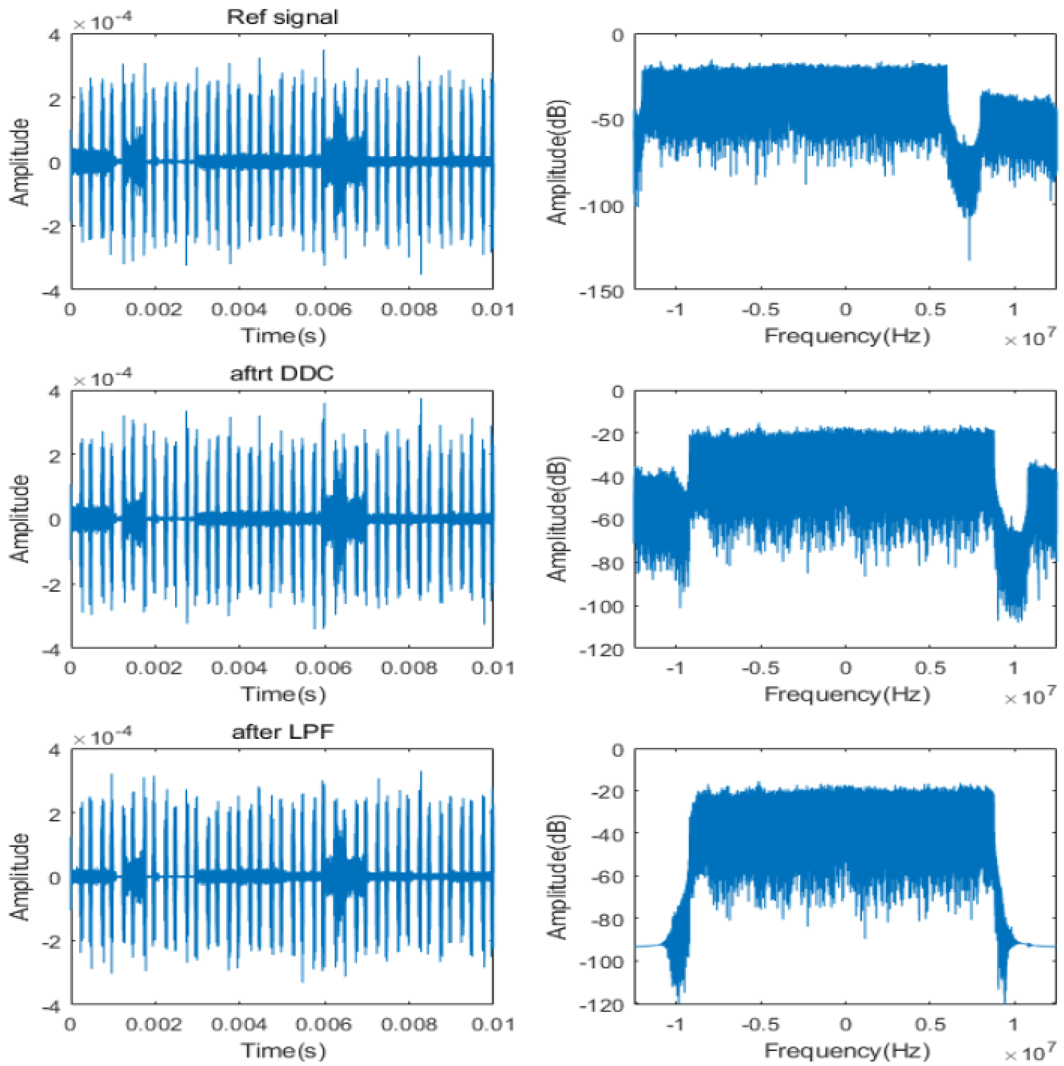

Figure 8 Signal processing process of the reference signal

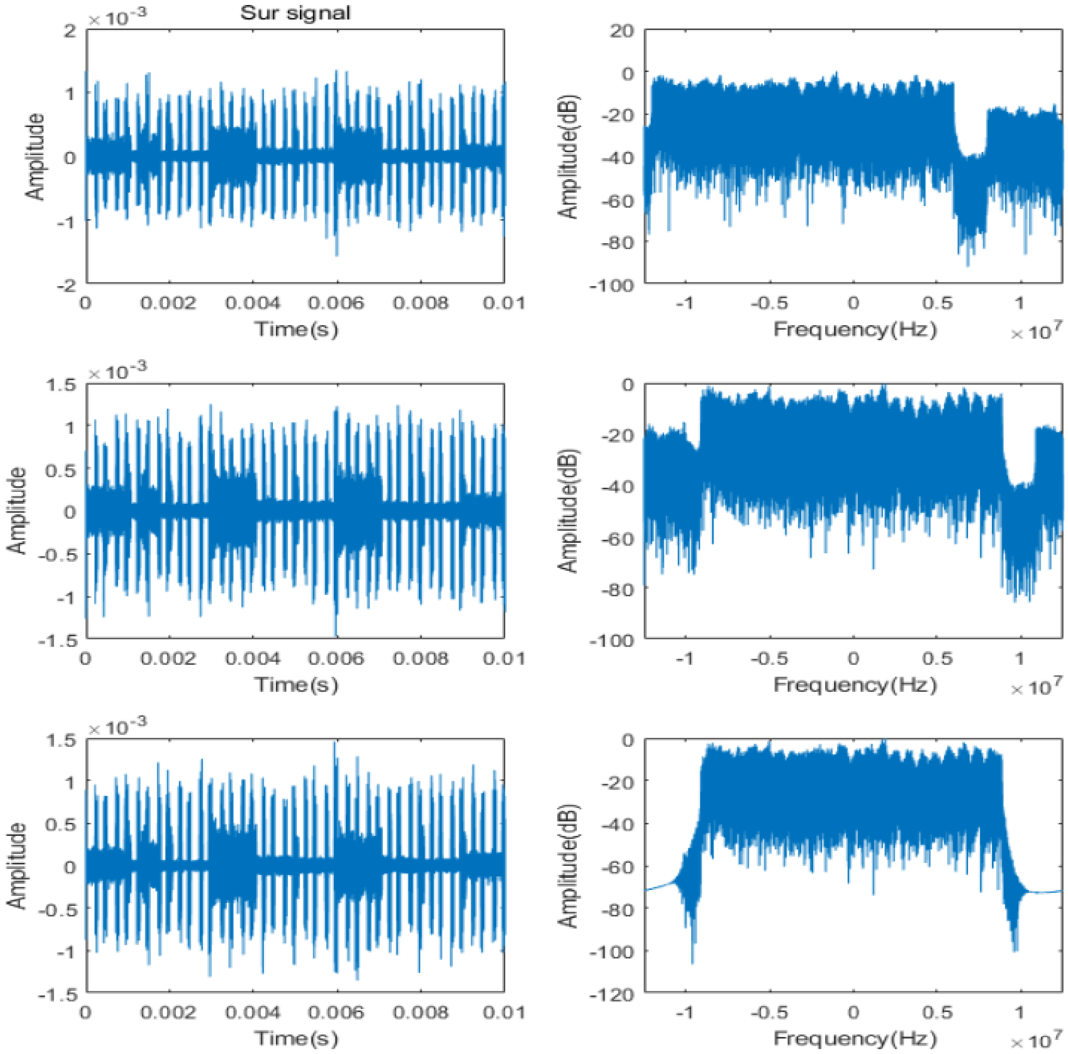

Figure 9 Signal processing process of the surveillance signal

Task 2. Drawing the Heatmaps

Step 1. Ambiguity Function Process

From the section Methodology we know how to calculate Δt and f_d. In this step we will implement this procedure.

For each data file we take 6 sampling points in time shift (0 ~ 5 sampling intervals) and 40 sampling points in frequency shift (-40 ~ 38 Hz). Then we iterate through all the above points and calculate the ambiguity function values.

```
function [tau,fD,cor_max,taus,fDs,cors] = cor(ysur,yref,Ts,tau_min,tau_max,Ntau,fD_min,fD_max,NfD)
yref_conj=conj(yref);
taus=linspace(tau_min,tau_max,Ntau);
Ns=floor(taus/Ts);
fDs=linspace(fD_min,fD_max,NfD);
N=length(ysur);
cors=zeros(Ntau,NfD);
for j=1:Ntau
    for k=1:NfD
        for l=1:N
            if((l-Ns(j))<1)
                yref_conj_current=0;
            else
                yref_conj_current=yref_conj(l-Ns(j));
            end
            cors(j,k)=cors(j,k)+ysur(l)*yref_conj_current*exp(-2i*pi*fDs(k)*(l-1)*Ts);
        end
    end
end
cor_max = max(max(cors));
[tau,fD] = find(cor_max == cors);
end
```

Figure 10 Ambiguity Function

Step 2. Plot the Heatmaps for The Ambiguity Function

Then we get the heatmap for this data file, and here's code implementation.

```
% Task 2 %
clc;

% Load
load data_15.mat

f0 = 2900000;
t = linspace(0,0.5,12500000);
sur_shift = seq_sur .* exp(1i*2*pi*f0.*t);
ref_shift = seq_ref .* exp(1i*2*pi*f0.*t);
[b,a] = butter(10,0.88*10^7*2/f_s,'low');
ref_shift_butter = filter(b,a,ref_shift);
sur_shift_butter = filter(b,a,sur_shift);

% 函数返回值:
% tau_max, fD_max, cor_max为最优解
% taus, fDs, cors为所有解

[tau_max,fD_max,cor_max,taus,fDs,cors] = cor(sur_shift_butter,ref_shift_butter,1/f_s,0,20*10^(-8),6,-40,40,41);
```

Figure 11 Code Implementation

```
% draw figure

range_s = taus*3*10^8;
fD_max_i = fDs(fD_max);
tau_max_i = taus(tau_max) * 10^8;

fprintf('Range Doppler Spectrum\n7s-7.5s f: %d tau: %1d', fD_max_i, tau_max_i);

figure;
imagesc(fDs, range_s, abs(cors));
colorbar;
```

Figure 12 Generate Heatmap

Here is the result:

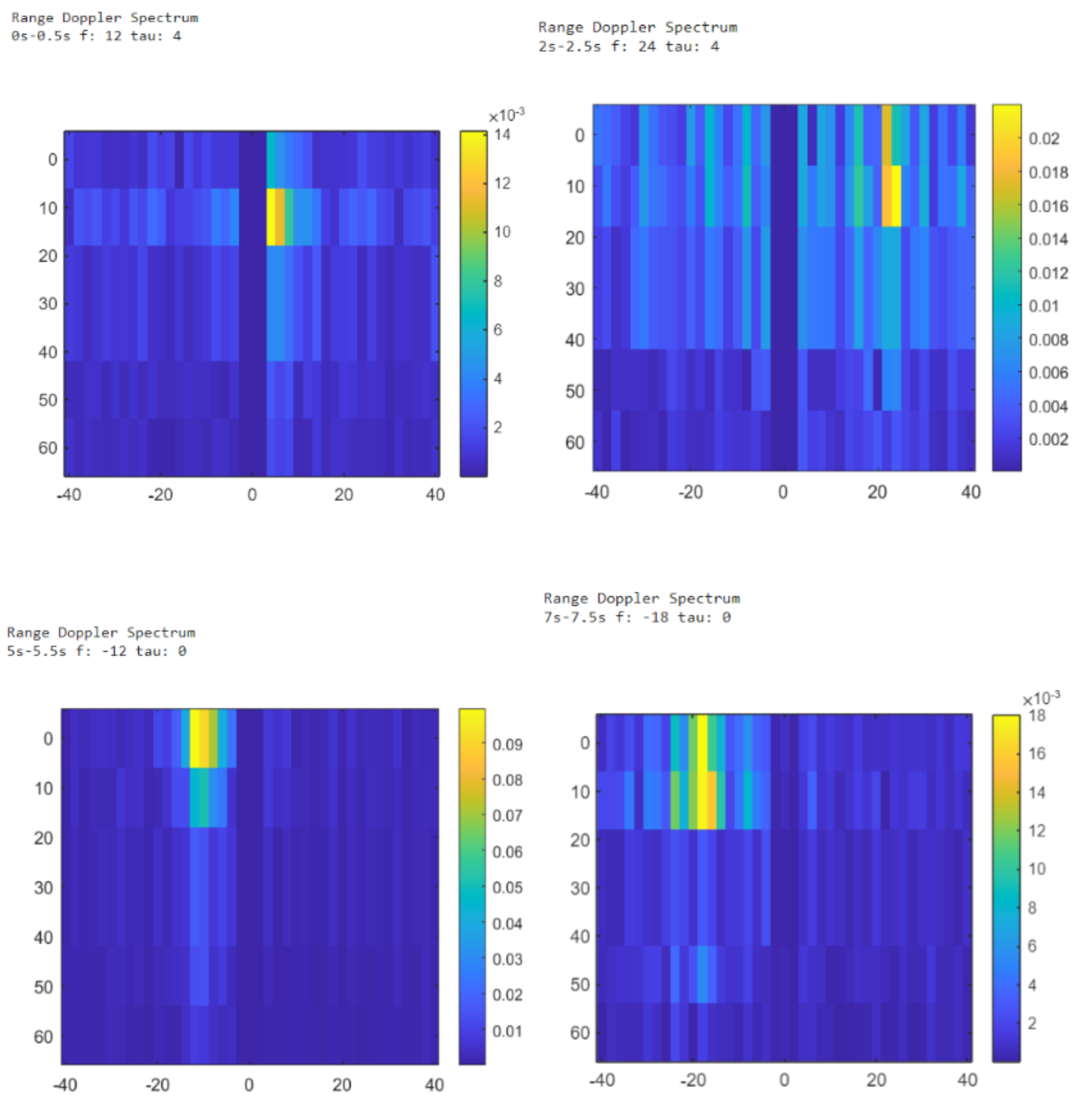

Figure 13 Heatmaps for data files

Step 3. Plot the Doppler Frequency – Time Relation

Then we selected more sampling points in timeline and plotted a big heatmap for the Doppler Frequency – Time Relation.

The time scale of this picture is to extract the result at the best Δt value every 0.5s, which shows the motion data of the target object to a certain extent.

```
% record

result(20, :) = cors(tau_max, :);
```

Figure 14 Abstract the Max

```
figure;
imagesc(fDs, 1:20, abs(result));
colorbar;
```

Figure 15 Generate Heatmap

Here is the result:

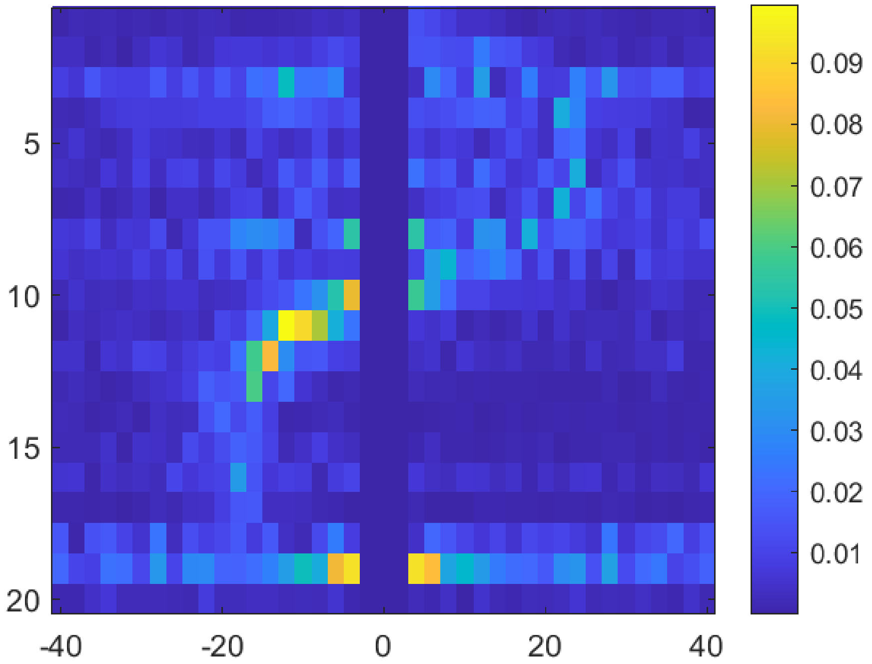

Figure 16 the Whole Heatmap

<Discussion>

Further Ideas

1. *The influence of Micro-Doppler effect in radar*

When, in addition to the constant Doppler frequency shift induced by the bulk motion of a radar target, the target or any structure on the target undergoes micro-motion dynamics, such as mechanical vibrations or rotations, the micro-motion dynamics induce Doppler modulations on the returned signal, referred to as the micro-Doppler effect.

The phase function can be expressed mathematically by introducing micro-motions to enhance traditional Doppler analysis.

The micro-Doppler effect can be used to identify object features. Specific applications include ground moving object detection, missile decoy recognition, human micro-motion detection, etc.

$$f_{\text{micro-Doppler}} = \frac{2f}{c}[\vec{\omega} \times \vec{r}]_{\text{radial}}$$

The above formula shows the calculation process of micro-Doppler frequency.

Problems During the Project

1. Code implementation of task requirements.
2. Code specification and code docking.
3. Be positive, be patient. T_T

What We Have Learned

1. Review how to use MATLAB functions to conduct signal processing.
2. Accumulated knowledge related to generate heatmaps, store files, etc., laying the foundation for future processing that may be required in similar projects.
3. Learn how to collect and refer to relevant materials, and further develop assumptions and discussions, construct models and corresponding explanations.

References

H. Kuschel, D. Cristallini and K. E. Olsen. (2019). "Tutorial: Passive radar tutorial," IEEE Aerospace and Electronic Systems Magazine. vol. 34 (2), 2-19, DOI: 10.1109/MAES.2018.160146.

V. C. Chen, F. Li, S. . -S. Ho and H. Wechsler. (2006), "Micro-Doppler effect in radar: phenomenon, model, and simulation study," in IEEE Transactions on Aerospace and Electronic Systems, vol. 42(1). 2-21. DOI: 10.1109/TAES.2006.1603402.

W. Dong, T. Sun, J. Tian, J. Wang, Z. Song and Q. Fu. (2022). "A Target Location Algorithm Based on Millimeter Wave Radar," 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC). 646-651, DOI: 10.1109/ITOEC53115.2022.9734670.

Appendix 1

```
% Task 1 %
clear;clc;


% Load
load data_1.mat


% 命名规则：
% ref: seq_ref 截取出的 0~0.01s 的信号
% ref_f: ref 的频域信号
% ref_shift: ref 经过频移后的时域信号
% ref_shift_f: ref_shift 的频域信号
% 以此类推...
```

```
% 画初始 ref,sur 信号的时域图


part = 250000; % 截取 0~0.01s 的数据
ref = seq_ref(1:part);
sur = seq_sur(1:part);
t = linspace(0,0.01,part);
figure;
subplot(3,4,1);
plot(t,abs(ref)),
xlabel('Time(s)'),
ylabel('Amplitude'),
title('Ref signal');


subplot(3,4,3);
plot(t,abs(sur)),
xlabel('Time(s)'),
ylabel('Amplitude'),
title('Sur signal');
```

```
% 画初始 ref,sur 信号的频域图


part = 250000;
ref_f = fft(ref);
tau = 0.01/part;
wk = (-pi/tau + (0:part-1) * 2*pi/part/tau)/2/pi;
subplot(3,4,2);
plot(wk,abs(20*log10(fftshift(ref_f)))),
xlabel('Frequency(Hz)'),
ylabel('Amplitude(dB)');
sur_f = fft(sur);
subplot(3,4,4);
plot(wk,abs(20*log10(fftshift(sur_f)))),
xlabel('Frequency(Hz)'),
ylabel('Amplitude(dB)');
```

```
% 画频移过后的 ref,sur 信号频域图


f0 = 2900000;
t = linspace(0,0.01,part);
sur_shift = sur .* exp(1i*2*pi*f0.*t);
subplot(3,4,8);
plot(wk,abs(20*log10(fftshift(fft(sur_shift))))),
xlabel('Frequency(Hz)'),
ylabel('Amplitude(dB)');
```

```
f0 = 2800000;
t = linspace(0,0.01,part);
ref_shift = ref .* exp(1i*2*pi*f0.*t);
subplot(3,4,6);
plot(wk,abs(20*log10(fftshift(fft(ref_shift))))),
xlabel('Frequency(Hz)'),
ylabel('Amplitude(dB)');
```

```
% 画频移过后的 ref,sur 信号时域图
```

```
subplot(3,4,5);
plot(t,abs(ref_shift)),
xlabel('Time(s)'),
ylabel('Amplitude'),
title('aftrt DDC');
```

```
subplot(3,4,7);
plot(t,abs(sur_shift)),
xlabel('Time(s)'),
ylabel('Amplitude');
```

```
% 画低通滤波后的 sur,ref 时域图


% LPF 低通的截止频率为 0.9*10^7
[b,a] = butter(10,0.88*10^7*2/f_s,'low');
ref_shift_butter = filter(b,a,ref_shift);
sur_shift_butter = filter(b,a,sur_shift);


subplot(3,4,9);
plot(t,abs(ref_shift_butter)),
xlabel('Time(s)'),
ylabel('Amplitude'),
title('after LPF');
```

```
subplot(3,4,11);
plot(t,abs(sur_shift_butter)),
xlabel('Time(s)'),
ylabel('Amplitude');
```

```
% 画低通后的 sur,ref 频域图


subplot(3,4,10);
```

```
plot(wk,abs(20*log10(fftshift(fft(ref_shift_butter))))),
xlabel('Frequency(Hz)'),
ylabel('Amplitude(dB)');
```

```
subplot(3,4,12);
plot(wk,abs(20*log10(fftshift(fft(sur_shift_butter))))),
xlabel('Frequency(Hz)'),
ylabel('Amplitude(dB)');
```

Appendix 2

```
% Task 2 %
clc;


% Load
load data_15.mat


f0 = 2900000;
t = linspace(0,0.5,12500000);
sur_shift = seq_sur .* exp(1i*2*pi*f0.*t);
ref_shift = seq_ref .* exp(1i*2*pi*f0.*t);
[b,a] = butter(10,0.88*10^7*2/f_s,'low');
ref_shift_butter = filter(b,a,ref_shift);
sur_shift_butter = filter(b,a,sur_shift);


% 函数返回值:
% tau_max, fD_max, cor_max 为最优解
% taus, fDs, cors 为所有解


[tau_max,fD_max,cor_max,taus,fDs,cors] =
cor(sur_shift_butter,ref_shift_butter,1/f_s,0,20*10^(-8),6,-40,40,41);
```

```
% draw figure


range_s = taus*3*10^8;
fD_max_i = fDs(fD_max);
tau_max_i = taus(tau_max) * 10^8;


fprintf('Range Doppler Spectrum\n7s-7.5s f: %d tau: %1d', fD_max_i,
tau_max_i);


figure;
```

```
imagesc(fDs, range_s, abs(cors));
colorbar;
```

```
function [tau,fD,cor_max,taus,fDs,cors] =
cor(ysur,yref,Ts,tau_min,tau_max,Ntau,fD_min,fD_max,NfD)
yref_conj=conj(yref);
taus=linspace(tau_min,tau_max,Ntau);
Ns=floor(taus/Ts);
fDs=linspace(fD_min,fD_max,NfD);
N=length(ysur);
cors=zeros(Ntau,NfD);
for j=1:Ntau
    for k=1:NfD
        for l=1:N
            if((l-Ns(j))<1)
                yref_conj_current=0;
            else
                yref_conj_current=yref_conj(l-Ns(j));
            end
            cors(j,k)=cors(j,k)+ysur(l)*yref_conj_current*exp(-
2i*pi*fDs(k)*(l-1)*Ts);
        end
    end
end
cor_max = max(max(cors));
[tau,fD] = find(cor_max == cors);
end
```

Appendix 3

```
% Task 3 %
clc;
result = zeros(20*41);


% Load
load data_20.mat


f0 = 2900000;
t = linspace(0,0.5,12500000);
sur_shift = seq_sur .* exp(1i*2*pi*f0.*t);
ref_shift = seq_ref .* exp(1i*2*pi*f0.*t);
[b,a] = butter(10,0.88*10^7*2/f_s,'low');
ref_shift_butter = filter(b,a,ref_shift);
sur_shift_butter = filter(b,a,sur_shift);


% 函数返回值:
% tau_max, fD_max, cor_max 为最优解
% taus, fDs, cors 为所有解


[tau_max,fD_max,cor_max,taus,fDs,cors] =
cor(sur_shift_butter,ref_shift_butter,1/f_s,0,20*10^(-8),6,-40,40,41);
```

```
% draw figure


range_s = taus*3*10^8;
fD_max_i = fDs(fD_max);
tau_max_i = range_s(tau_max);


fprintf('Range Doppler Spectrum\n0s-0.5s f: %d tau: %d', tau_max,
fD_max);
```

```
figure;
imagesc(fDs, range_s, abs(cors));
colorbar;
```

```
% record


result(20, :) = cors(tau_max, :);
```

```
figure;
imagesc(fDs, 1:20, abs(result));
colorbar;
```

```
function [tau,fD,cor_max,taus,fDs,cors] =
cor(ysur,yref,Ts,tau_min,tau_max,Ntau,fD_min,fD_max,NfD)
yref_conj=conj(yref);
taus=linspace(tau_min,tau_max,Ntau);
Ns=floor(taus/Ts);
fDs=linspace(fD_min,fD_max,NfD);
N=length(ysur);
cors=zeros(Ntau,NfD);
for j=1:Ntau
    for k=1:NfD
        for l=1:N
            if((l-Ns(j))<1)
                yref_conj_current=0;
            else
                yref_conj_current=yref_conj(l-Ns(j));
            end
            cors(j,k)=cors(j,k)+ysur(l)*yref_conj_current*exp(-
2i*pi*fDs(k)*(l-1)*Ts);
        end
    end
end
cor_max = max(max(cors));
[tau,fD] = find(cor_max == cors);
end
```