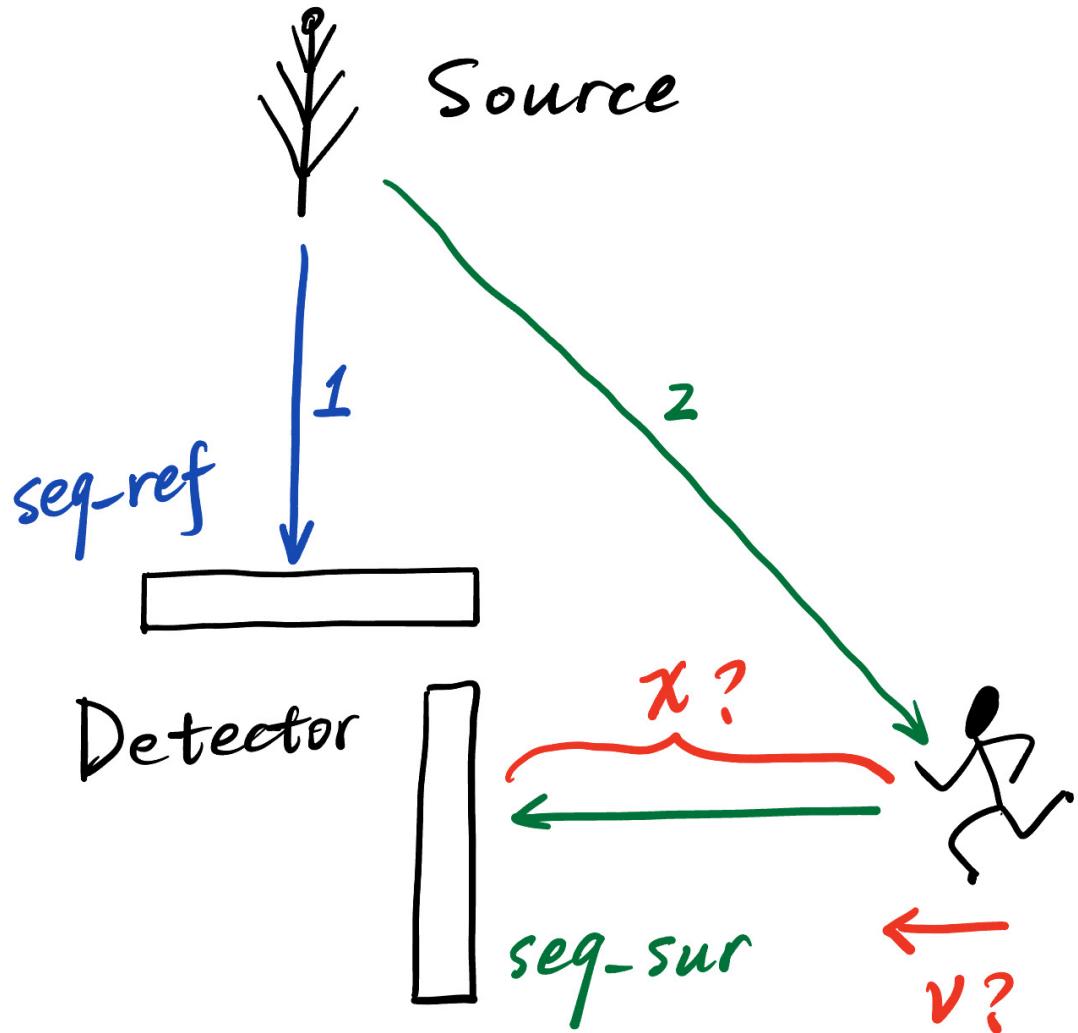


# 基于通信信号的运动检测

PROJECT 2

12212859 黎思婕

# 检测原理



| 位置 $x$                    | 速度 $v$                  |
|---------------------------|-------------------------|
| $x = c\tau$<br>近似 $1 = 2$ | $v = \frac{f_d c}{f_c}$ |

c 光速  
 $f_c$  载波中心频率  
2123MHz

$f_d$  多普勒频率

$\tau$  时间差

已知

利用模糊函数寻找

# TASK1 数据处理

## 目标

对20个0.5秒的 `seq_ref` 和 `seq_sur` 进行操作，使得能够带入模糊函数

## 已知

|                                |         |
|--------------------------------|---------|
| <code>f_c</code> 载波中心频率        | 2123MHz |
| <code>f_s</code> 采样率           | 25MHz   |
| <code>duration (T)</code> 片段时间 | 0.5s    |
| <code>f_ddc</code> 频率平移量       | -3MHz   |
| <code>bandwidth</code> 目标信号带宽  | 9MHz    |

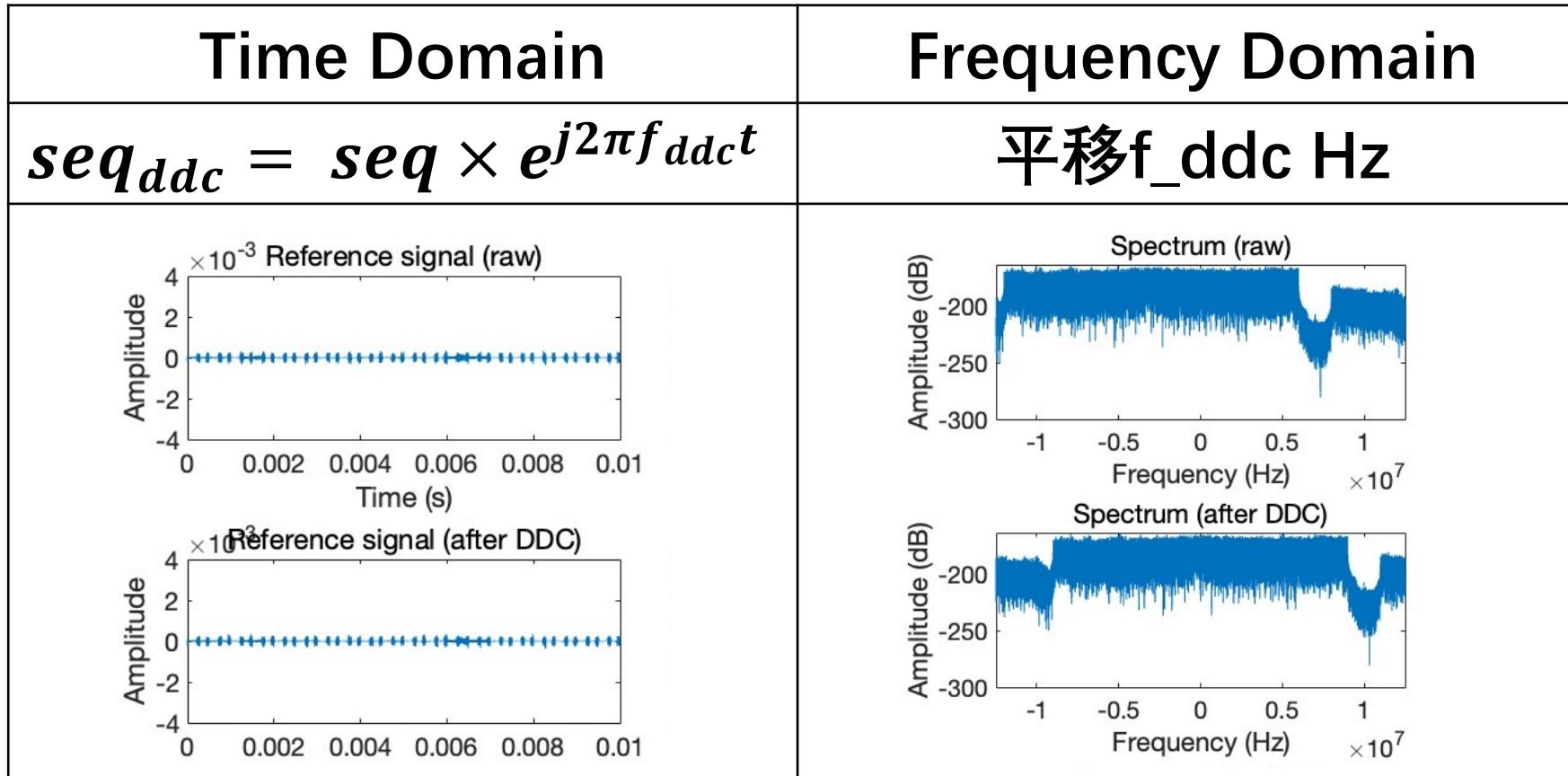
$$\text{采样间隔 } \tau_0 = \frac{1}{f_s}$$

$$\text{片段样本数量 } N = \frac{T}{\tau_0} = 0.5f_s$$

# TASK1 数据处理

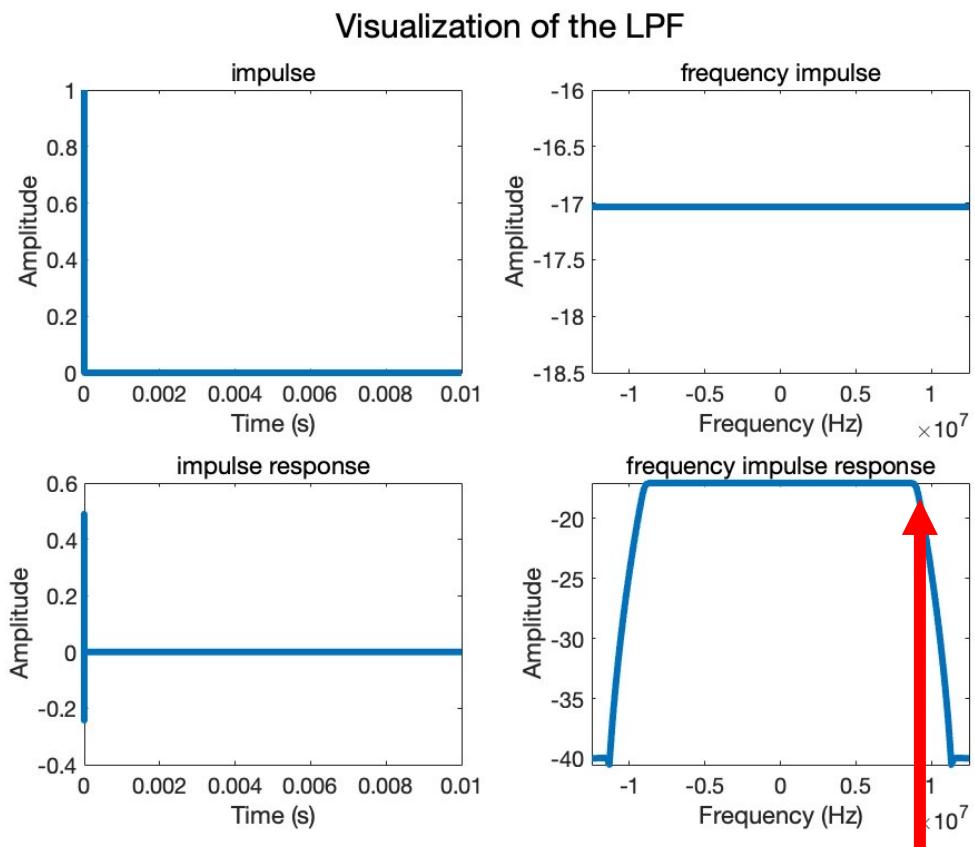
## Step1 Load Data

## Step2 DDC



# TASK1 数据处理

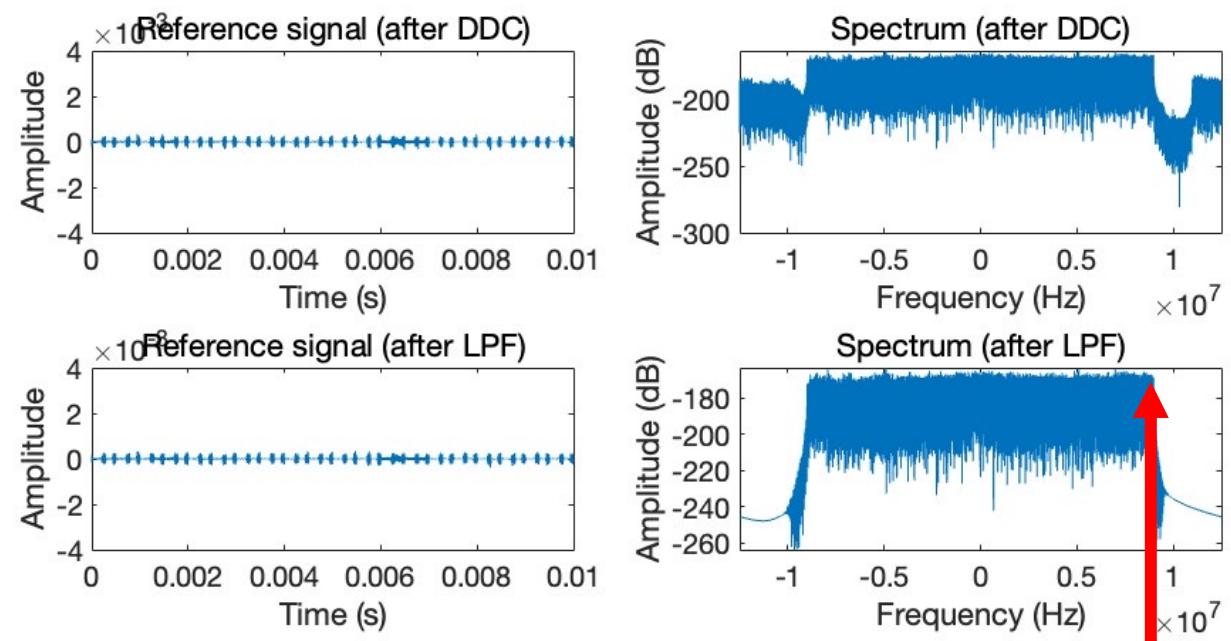
## Step3 LPF



9MHz

### Design of LPF

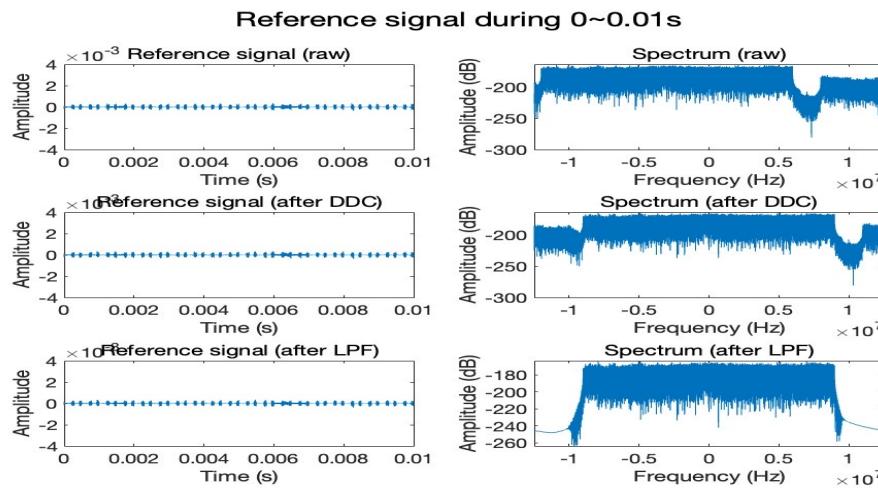
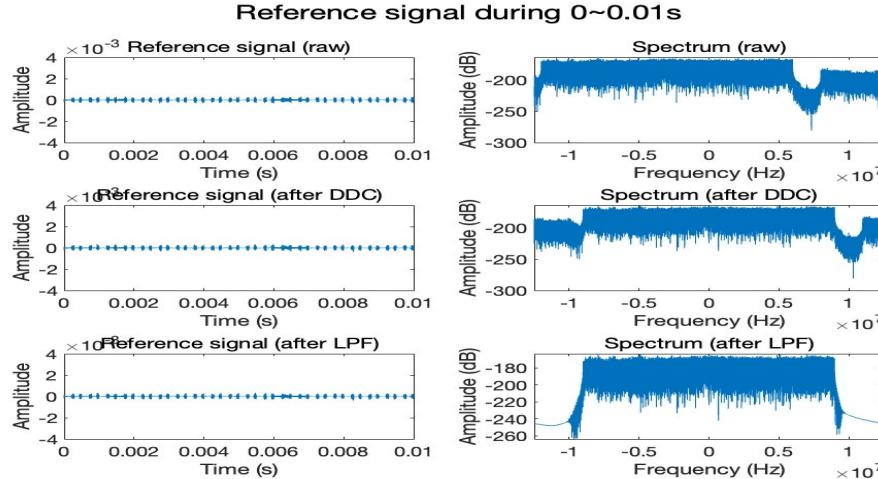
```
[b,a]=butter(20,bandwidth/(f_s/2))  
seq_lpf=filter(b,a,seq_ddc)
```



9MHz

# TASK1 数据处理

## Result



## 关键细节

- 画0 ~ 0.01s部分，只处理这么长时间的来做图，因为傅立叶变换后无法对时间操作，无法取0 ~ 0.01s对应的频率成分做图。
- 只对时域操作，频域仅作理解与展示
- FFT过程（所有作图均用`fftshift`）  
`X_ref=fftshift(tau.*fft(seq_ref))`
- 横坐标取物理频率  
 $w=2*pi/N/tau*(-N/2:N-1-N/2); f=w./(2*pi)$
- 频域信号强度单位取dB  
`plot(f, 20.*log10(abs(X_ref_ddc)))`
- 时域取实部作图（因为原信号经处理为复数）  
`plot(t, real(seq_ref_lpf))`

# TASK2 距离- $f_D$ 图

TASK1 数据处理

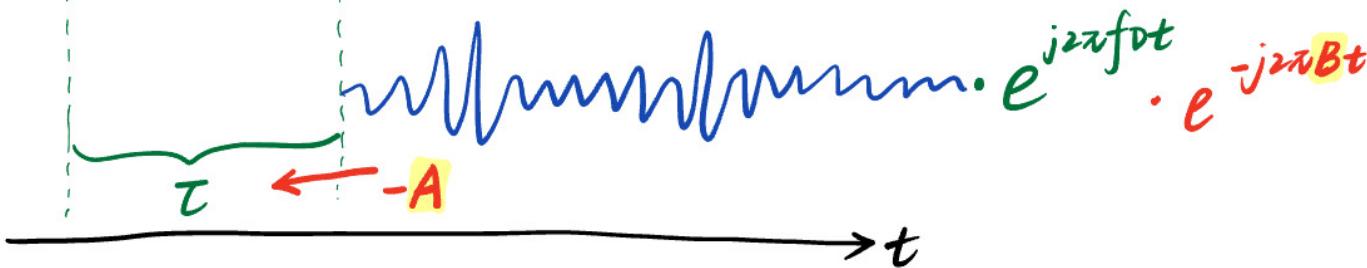
模糊函数

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

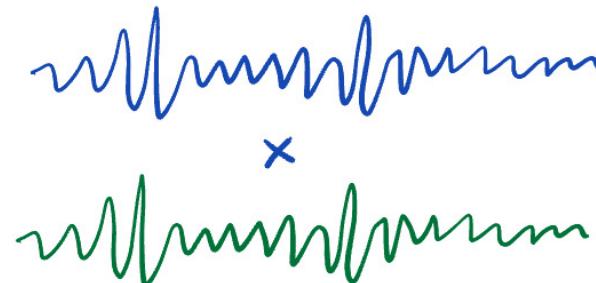
seq\_ref



seq\_sur



当  $A = \tau$  (时间差)  
 $B = f_d$  (多普勒频率)



$$= \left( \text{amplitude of } \text{seq\_sur} \right)^2$$

**MAX**

# TASK2 距离-f<sub>D</sub>图

TASK1 数据处理

## 模糊函数

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

## For Loop

```
n_tau=0:6;
range=n_tau*tau*3e8;
cor=zeros([length(n_tau),length(f_d)]);
for k=n_tau. % 遍历A
    for m=1:length(f_d) % 遍历B
        for n=1:N
            if n-k>0
                cor(k+1,m)=cor(k+1,m)+ seq_sur_lpf(n)*conj(seq_ref_lpf(n-
k))*exp(-1j*2*pi*f_d(m)*(n-1)*tau);
            end
        end
    end
end
```

# TASK2 距离-f<sub>D</sub>图

## TASK1 数据处理

### For Loop

```
n_tau=0:6;
range=n_tau*tau*3e8;
cor=zeros([length(n_tau),length(f_d)]);
for k=n_tau. % 遍历A
    for m=1:length(f_d) % 遍历B
        for n=1:N
            if n-k>0
                cor(k+1,m)=cor(k+1,m)+ seq_sur_lpf(n)*conj(seq_ref_lpf(n-k))*exp(-1j*2*pi*f_d(m)*(n-1)*tau);
            end
        end
    end
end
```

## 确定遍历值

### Range

- 精度12m
- 人运动范围0~72m

$$dx = c \times d\tau$$

$$d\tau_{\{min\}} = \tau_0 = \frac{1}{f_s}$$
$$dx_{\{min\}} = 12m$$

### Doppler Frequency

- 人运动速度范围 ±5.65m/s  
得出-40Hz~40Hz

$$\nu = \frac{f_d c}{f_c}$$

# TASK2 距离-f<sub>D</sub>图

TASK1 数据处理

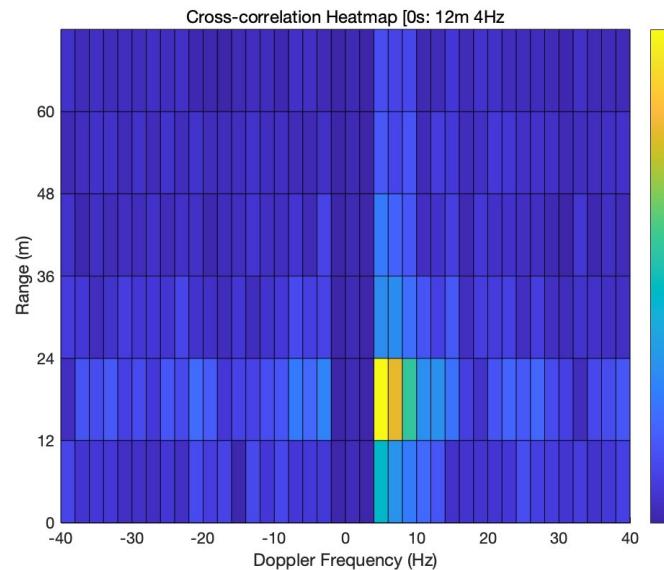
## 增加作图对比度

% 初步缩小范围

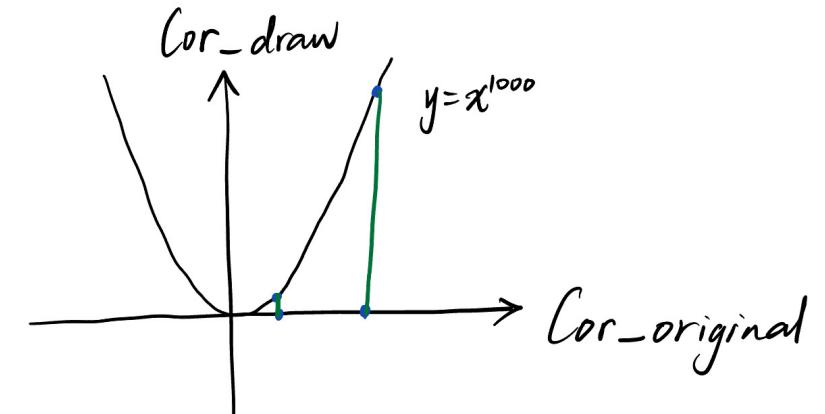
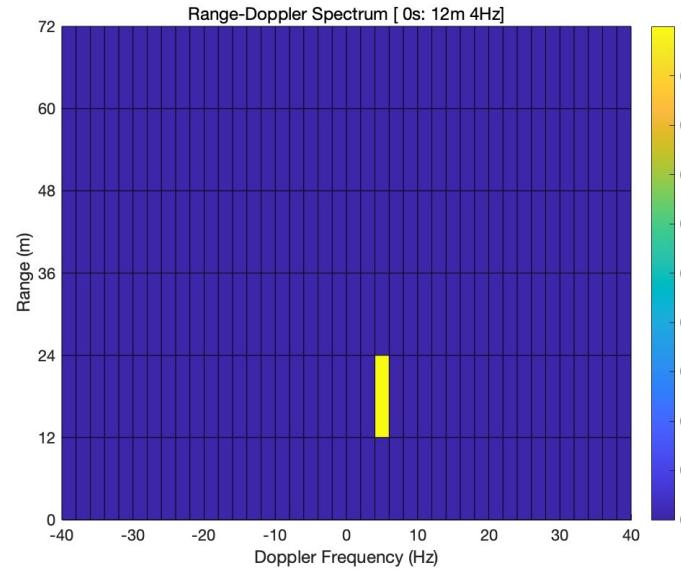
```
Cor = Cor / max(Cor(:));
```

% 使用幂指数函数进行非线性映射把小的值变小，大的值变大

```
Cor=(Cor).^1000;
```



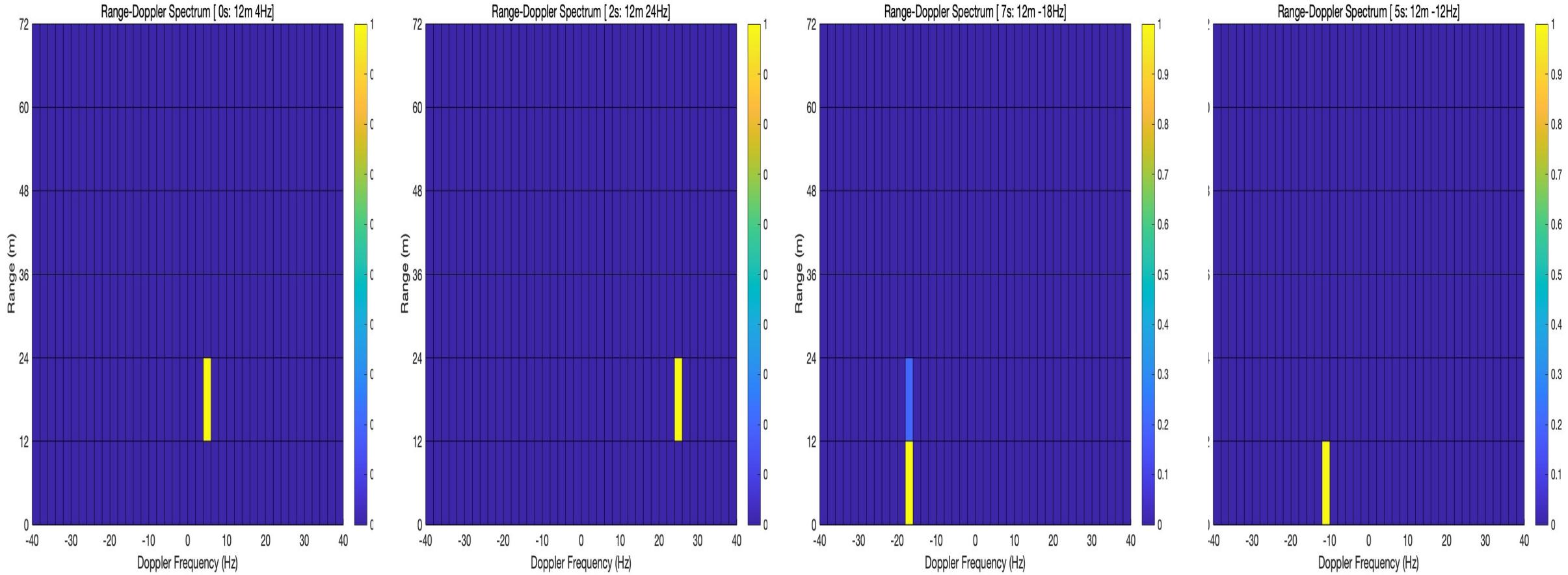
VS



# TASK2 距离- $f_D$ 图

TASK1 数据处理

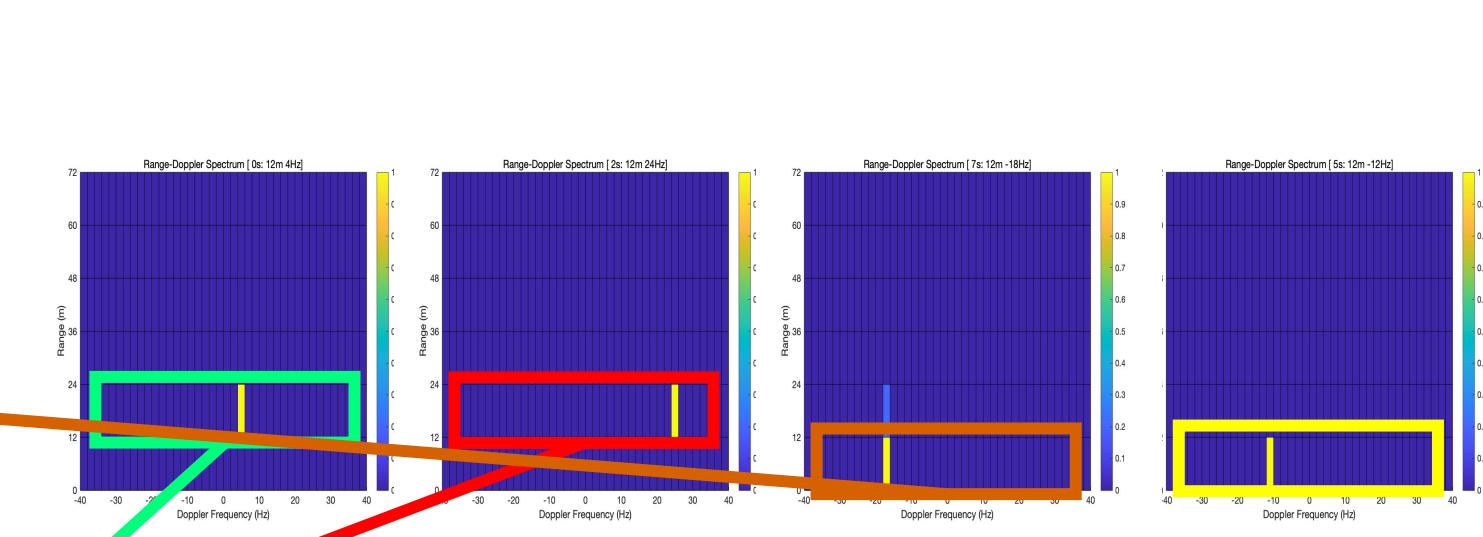
## Result



# TASK3 时间-f<sub>D</sub>图

TASK1 数据处理

TASK2 距离-fD图



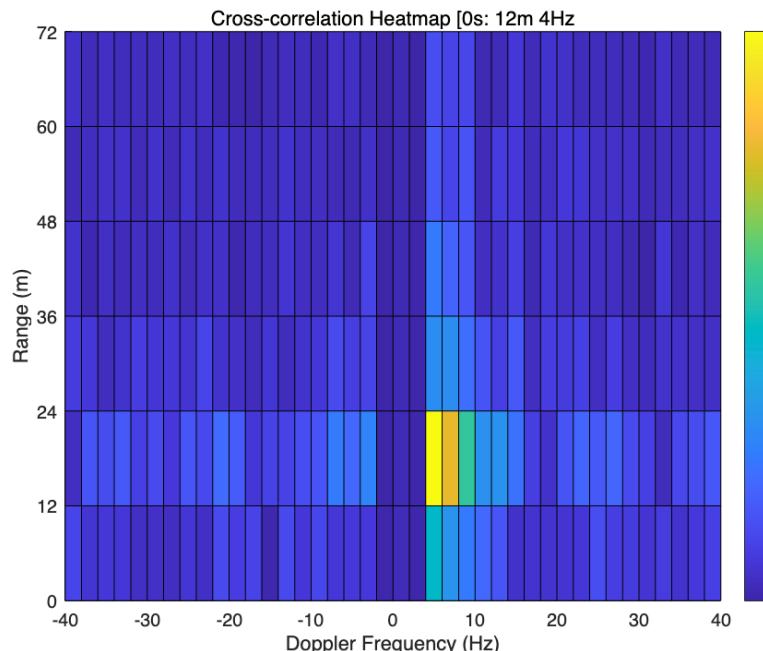
取最大值所在行粘起来

# 拓展-利用FFT提高速度

TASK1 数据处理

TASK2 距离-f<sub>D</sub>图

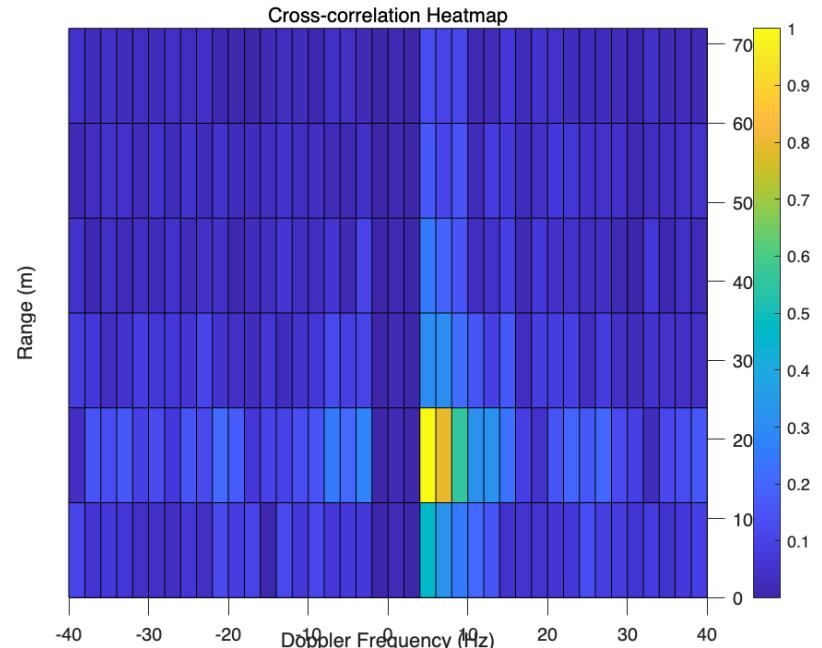
TASK3 时间-f<sub>D</sub>图



历时 233.891536 秒。

233.9秒

VS



历时 6.615411 秒。

6.6秒

# 拓展-利用FFT提高速度

TASK1 数据处理

233.9秒

TASK2 距离- $f_D$ 图



TASK3 时间- $f_D$ 图

6.6秒

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

$$\begin{aligned} & \left\{ \sum_k \left( y_{surv}[k] e^{-j\frac{\omega}{2\pi B} k} \right) \cdot y_{ref}^*[A-k] \right\} \\ &= Y_{surv}(e^{j\omega + \frac{\omega}{2\pi B}}) \cdot Y_{ref}^*(e^{j\omega}) \end{aligned}$$

## Code Realization

```
a_ref=conj(fft(flipud(seq_ref_lpf)));
b_sur=fft(seq_sur_lpf);
Cor=[];
for k=-20:20
    Cor_fft=a_ref.*circshift(b_sur,-k);
    Cor_k=ifft(Cor_fft);
    Cor=cat(2,Cor,Cor_k(1:7).');
end
```

# 拓展-多维精度分辨率提高

TASK1 数据处理

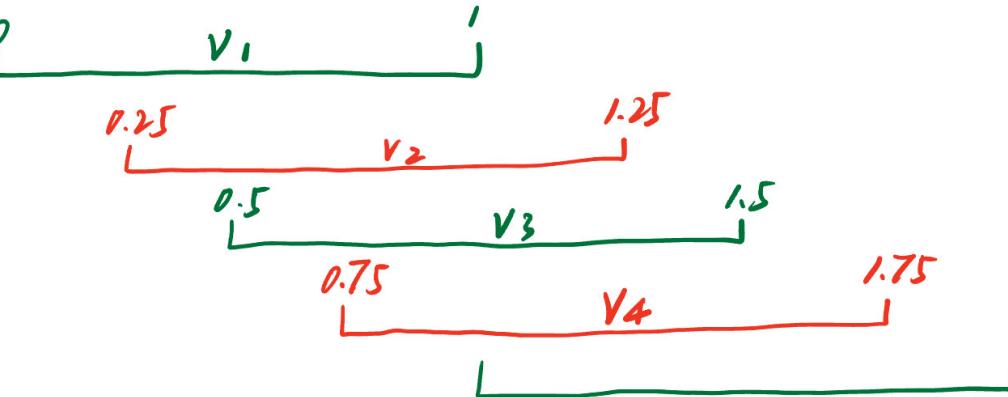
TASK2 距离-f<sub>D</sub>图

TASK3 时间-f<sub>D</sub>图

time(s) 0 0.25 0.5 0.75 1 1.25 1.5 1.75 2 ...

origin 

sliding window  
size=1s  
step=0.25s



0 0.25 0.5 0.75 1 1.25 1.5 1.75 2 ...

v<sub>1</sub> v<sub>2</sub> v<sub>3</sub> v<sub>4</sub> ...

0 0.25 0.5 0.75 1 1.25 1.5 1.75 2 ...

v<sub>1</sub> v<sub>2</sub> v<sub>3</sub> v<sub>4</sub> ...

## 1.时间精度-step

0.0 | 0.5 | 1.0...  
0.00 | 0.25 | 0.50 | 0.75 | 1.00 ...

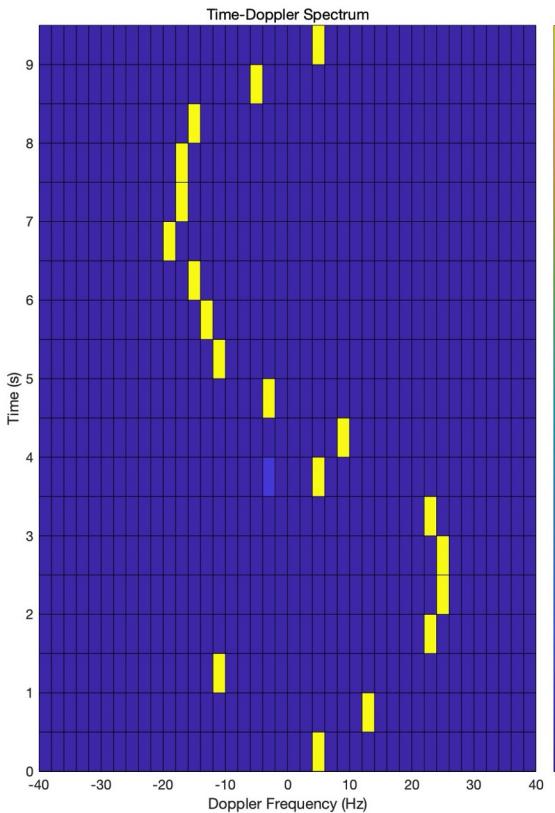
## 2.分辨率-size

0 0.25 0.5 0.75  
  
0 0.25 0.5 0.75  
v<sub>1</sub> v<sub>2</sub> ...

更长的时间得到的平均速度更准确  
关注动态变化

# 拓展-多维精度分辨率提高

TASK1 数据处理



TASK2 距离-f<sub>D</sub>图

## 3. 频率精度

$$\omega_0 = 2\pi (\Delta f_d)_{\{min\}} = \frac{2\pi}{N\tau_0}$$

$$\tau_0 = \frac{1}{f_s}, N = \frac{\text{duration}}{\tau_0} \quad (\text{片段样本数})$$

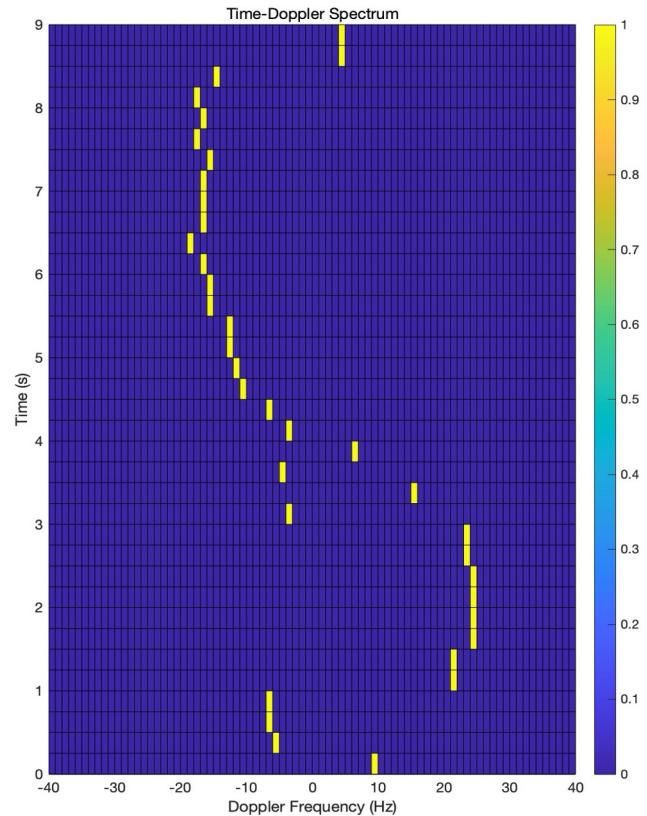
$$N \rightarrow 2N$$

$$\Delta f_d = 2\text{Hz} \rightarrow \frac{\Delta f_d}{2} = 1\text{Hz}$$

时间精度: 0.5s

速度精度: 0.282m/s

TASK3 时间-f<sub>D</sub>图

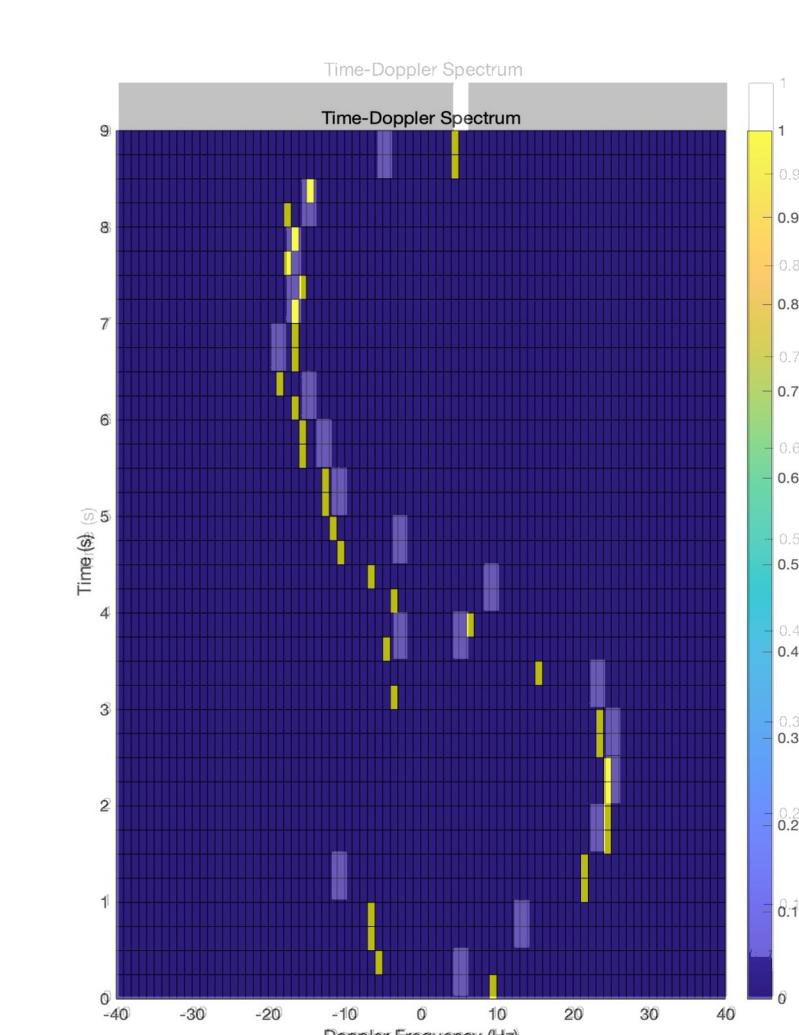


时间精度: 0.25s

速度精度: 0.141m/s

# 拓展-多维精度分辨率提高

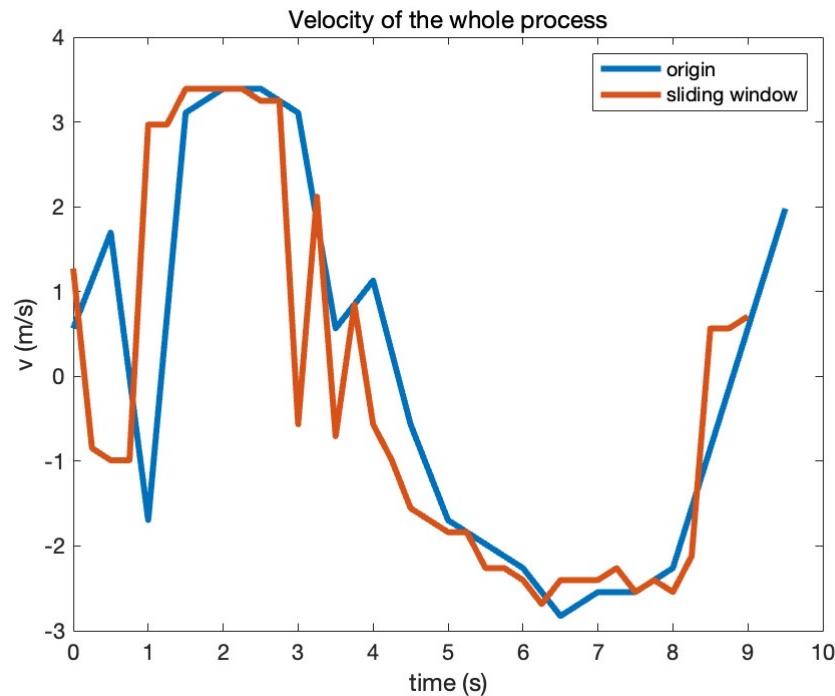
TASK1 数据处理



TASK2 距离- $f_D$ 图

(This section is empty)

TASK3 时间- $f_D$ 图



- **不重合**: 分辨率改变, 用更长时间算得的平均速度与短时间的不同
- **趋势相同**, 变化更精细

# 总结

## 基于通信信号的运动检测

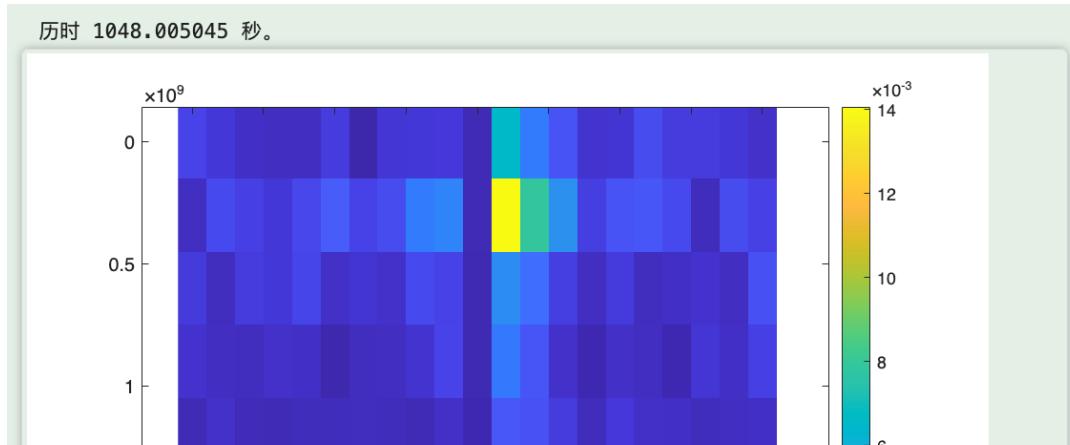
TASK1 数据处理

TASK2 距离- $f_D$ 图

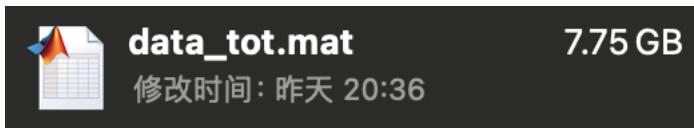
TASK3 时间- $f_D$ 图

## Failure

历时 1048.005045 秒。



## Computation

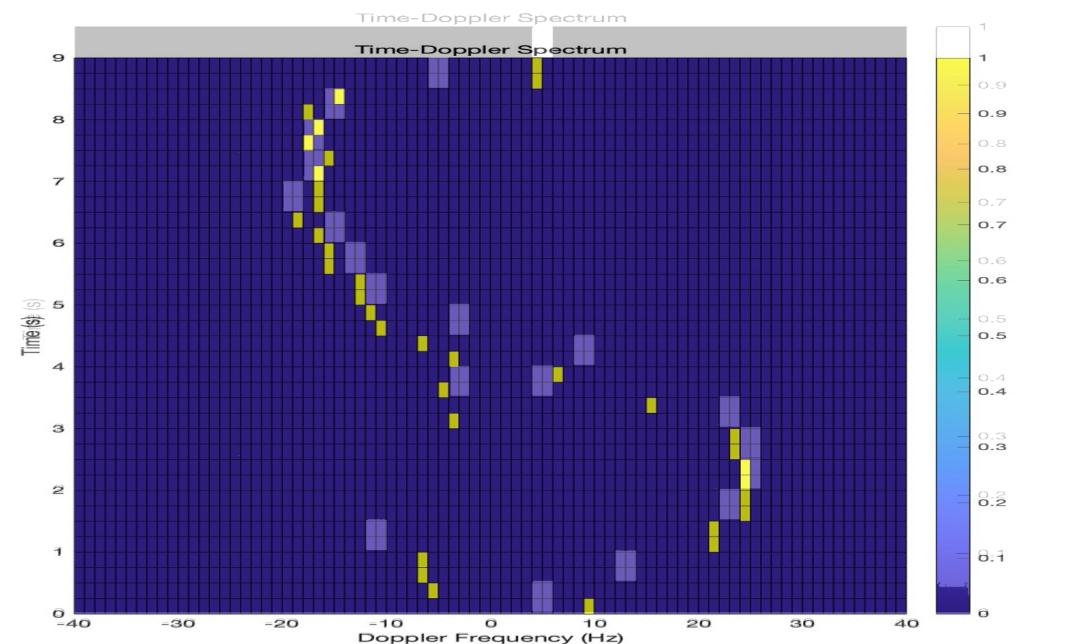


警告：未保存变量 'data\_ref\_lpf'。对于大于 2GB 的变量，请使用 MAT 文件版本 7.3 或更高版本。

警告：未保存变量 'data\_sur\_lpf'。对于大于 2GB 的变量，请使用 MAT 文件版本 7.3 或更高版本。

## Further Thinking

- 平衡计算量与步长
- 平衡分辨率（窗长）和动态信息量



# 谢谢

12212859 黎思婕  
12212860 黄姝颜  
12212861 魏雨杉