**Project 2: Motion Detection**

|  |  |
| --- | --- |
| **Author** | Name：闵泓硕，聂宇康，陈若翀，王泽奇  Student ID: 12212632，12111115，12112301，12011810 |
| **Introduction**   1. **Principles of Motion Detection**   Moving objects reflects radiation and causes time delay due to distance difference and frequency shift due to Doppler Effect of the radiation signal. The received signal without reflection is called reference signal, and the reflected one is called surveillance signal. We can analyze the two signals and find the time delay and frequency shift caused by the object and thus deriving the motion of it. Assume the reference signal to be and surveillance signal to be The two signal is given by  Where refers to the signal launched by the radar. stand for the attenuation factor due to transmission. Doppler Effect and relation of distance show that  where stands for the time delay, stands for the Doppler frequency shift (we call it Doppler shift in the following context), and indicates the angle between the radar, receiver and the moving object. To simplify the model, we assume . Then we derive  Ambiguity function is defined to estimate the relation of two signals, say and . The expression of the function is  Where and are variables. The value indicates how much the two signals overlap when one of the signals is time-delayed and frequency shifted. Especially, when  The two signals have the most coincidence (they overlap the most). And the value of the function reaches the maximum value. At the maximum point, the coordinates are identical to the real time delay and Doppler shift.  At each time , the object has a set of such “coordinates” , and such coordinates correspond to the current motion state at . Therefore, the ambiguity function is actually associated with the time as well: Thus, we need to find the coordinate set at each t. we firstly solve the maximum value of :  Then we plot the function as a vector in MATLAB.   1. **Digital Down Convert (DDC)**   Signals that carry information launched by radar are usually modulated by a sinusoidal function with a fairly high frequency (roughly 2000MHz). When sampling the signal without and process, the sampling rate needs to be critically high, which is very hard and will cause accuracy loss. Therefore, before sampling, we use digital down convertor to shift the signal in frequency domain to the base frequency band (narrow the frequency band).  In order to perform DDC, we firstly modulate the signal with a complex exponential periodic function, and the frequency is identical to the center of the original signal in frequency domain.  In Fourier Transform, we have  Then after sampling, we use a lowpass filter with its bandwidth the same as the frequency band of to block the other modulation signals, and the base part is left. The whole process is digital down convert.   1. **Digital Filter**   We have mentioned digital filter in part 2. The difference between digital filter and other filters is that digital filter is applied after sampling. That is to say, the digital filter is focused on the numeral operations rather than the convolutions of traditional filters. However, in MATLAB, the digital filters and traditional filters are all simulated by discrete, numeral signals. Thus, the design, verification and use are the same as the filters we use before.  One example is butterworth filter. In order to design a butterworth digital filter, we use:  [b, a] = butter(n, Wn)  Where n stands for the order of the filter, and Wn, which lies between 0 and 1, represents the frequency interval lies within fs\*Wn. And fs is the sample frequency. The “3dB” verification is also the same as before:  If the condition above is satisfied, the digital filter passes the test and can be used.   1. **Frequency Processing and Lowpass Filters Design**   In this lab, we not only use Fourier Transform to analyze the center frequency and frequency band of the received signal, but use modulation to conduct digital down convert before sampling as well. The modulation in MATLAB is:  x1=x0.\*exp(-1j\*2\*pi\*f\_ddc\*[0:duration\*f\_s-1]/f\_s);  The lowpass filter design is the same as former projects. The commands are the same as part3, and verification condition “3dB” rule applies to the lowpass filters as well.   1. **Possible Applications of Motion Detection**  * Security systems: Motion detection via radar can be used to detect intruders or suspicious activity in a restricted area, triggering alarms or alerting security personnel. * Traffic monitoring: Radar-based motion detection can be used to monitor traffic flow and detect accidents or congestion on roads and highways. * Industrial automation: Radar sensors can be used to monitor the movement of equipment and machinery in industrial settings, triggering alerts or shutting down machinery if motion is detected in restricted areas. * Robotics: Motion detection via radar can be used in robotics to detect the presence of objects or obstacles in the robot's path, allowing it to navigate around them.   **Lab results & Analysis**：  Task 1. Question:  Plot the waveform in time domain and frequency spectra of reference signal and surveillance signal in original state, after DDC and after LPF respectively.  Results & Analysis:  The original Reference signal and Surveillance signal are shown in the figures below.  Figure 1. Time Waveform and Frequency Spectrum of Original Reference Signal  Figure 2. Time Waveform and Frequency Spectrum of Original Surveillance Signal  The original signal consists of baseband signals in two frequency bands, 2110-2130MHz and 2130-2135MHz.According to the reference, we choose the 2110-2130 MHz (20M bandwidth) signal to do the motion detection. Therefore, we should convert this signal to zero intermediate frequency (IF), and filter out the high-frequency signal through the low-pass filter to complete the preliminary signal processing.  From the figure of original signal, we can find that we need to shift the whole signal to the right 3MHz in the frequency domain to shift the center of 2110-2130MHz signal to the zero IF.  In order to achieve this purpose, we can apply digital-down-conversion technology, multiplying the time domain signal by , while . The result of digital-down-conversion (DDC) processing is shown below.  Figure 3. Time Waveform and Frequency Spectrum of Reference Signal after DDC  Figure 4. Time Waveform and Frequency Spectrum of Surveillance Signal after DDC  Then we can construct a low-pass filter with a cutoff frequency of 9MHz to get the signal we want. The lowpass filter is shown in the figure, which meets the requirements.  Figure 5. Verification of Lowpass Filter  After passing the signal (after DDC) through this low-pass filter, we derive the desired signal:  Figure 6. Time Waveform and Frequency Spectrum of Filtered Reference Signal after DDC  Figure 7. Time Waveform and Frequency Spectrum of Filtered Surveillance Signal after DDC  Then we can compute the ambiguity function of it to get its Doppler frequency shift, so as to complete the motion detection.  The results of all initial signal processing are shown below.  Figure 8. Time Waveform and Frequency Spectrum of Reference Signal in All Three States  Figure 9. Time Waveform and Frequency Spectrum of Surveillance Signal in All Three States  Task 2. Question: Obtain the Range-Frequency Doppler plot of processed signal in time interval of 0-0.5s、2-2.5s、5-5.5s、7-7.5s respectively.  Results & Analysis:  Figure 10. Doppler Range-Frequency Plot at 0-0.5s  Figure 11. Doppler Range-Frequency Plot at 2-2.5s  Figure 12. Doppler Range-Frequency Plot at 5-5.5s    Figure 13. Doppler Range-Frequency Plot at 7-7.5s  Analysis：Firstly, four range-Doppler pictures are drawn. From these pictures, we can see that after obtaining the effective information of the two signals obtained in Task1, by analyzing the relationship between distance and Doppler in these four pictures, we can roughly infer the position of the exerciser at this time and roughly calculate the relative motion speed of the exerciser from the frequency value of the Doppler shift. But we can't see a specific trajectory yet.  Task 3. Question:  Derive the Frequency-time Doppler spectrum of 10 seconds of data processed with a sliding window of CIT=0.5s and sliding by 0.5s each time.  Results & Analysis:  Figure 14. Time-Frequency Doppler Plot at Fixed Distance with 0.5s sliding window  Extension: Plot the time-frequency Doppler Plot with a sliding window of CIT=0.1s  Figure 15. Time-Frequency Doppler Plot at Fixed Distance with 0.1s sliding window  Analysis：  An important and improved point in the experiment is the use of fft(), because we can convert the fuzzy function to the product of two Fourier transforms according to the convolution theorem. Thus, the number of multiplication operations is reduced, and the program speed is increased.  By taking out the value of the maximum ambiguity function from 20 distance Doppler spectra, and then forming a new graph, and changing the time interval from 0.5s to 0.1s, the last relatively accurate picture is obtained. From the picture, it can be roughly analyzed that the motion of the object should be away first and then close as the yellow trajectory indicates, and the frequency shift is firstly positive and then negative.  It shows that the direction of motion is changing, and the frequency shift at the end and the beginning is roughly zero, which coincides with the phenomenon of the velocity being almost zero before and after the beginning of motion. And the speed is almost constant when it is far away, but there is a more obvious change in the speed when it is near. Because the figure is straighter when the frequency shift is negative, and there is a significant bend when the frequency shift is positive, which indicates that the speed is changing fast.  Part 4. Applications of Motion Detection  In terms of application, motion detection technology based on communication signals, people can use common Wi-Fi signals to carry out passive personnel motion detection. Using the human body's influence on Wi-Fi signals to detect and track people in motion, this technology does not require the target to carry any signal transceiver equipment, nor does it require additional hardware to be installed in the monitoring environment. This technology has great application prospects in smart home, disaster relief, enterprise security and other fields.  In addition, motion signal detection technology is also often used in aviation rescue, further ensuring people's life safety. This technology is very promising. | |
| **Experience:**  12212632 闵泓硕：  This lab has provided me with valuable insights into the possible applications of the knowledge I have gained in Signals and Systems. The lab introduced the concept of ambiguity function and applied the theorem of Doppler frequency shift, integrating many new methods in MATLAB, including Digital Down Convert, digital filters, and PlotRD plotting tool. All these ideas and tools came together to achieve a surprising result of motion detection simulation.  To be honest, I never thought that motion detection could be realized by simply using Fourier Transform and some basic physics knowledge. When the Doppler Plots finally appeared in our computer screen, I still feel a little surprised that we can analyze the motion with given data indeed! If the previous labs were exercises, and project 1 was a comprehensible application of Fourier Transform, project 2 was more complicated, detailed, and interesting. Working in a group, I felt like a real engineering team.  On the other hand, the final project was more challenging than ever before. It took me about three days to understand the principle of detection and the concept of ambiguity function. Initially, I was confused about Digital Down Convert, but after learning modulation and demodulation comprehensively, I understood that DDC is another way of demodulating the signal carrying information modulated by a fairly high sinusoidal function. Drawing the figures and realizing them in MATLAB also took me a long time to grasp.  In summary, although this project was challenging, I learned a great deal from it and understood how to work as a team to solve realistic problems.  12111115 聂宇康：  Through this experiment, I once again applied digital-down-conversion, Butterworth filter and other techniques, which made me more proficient in signal processing and the use of MATLAB. At the same time, I learned how to use Riemann SUM to estimate integrals, how to use fuzzy functions to resolve multiple targets in a signal, and how to use Doppler shifts to estimate changes in motion. Although I was not very proficient in using this new knowledge, they broadened my horizons and told me what I could do with what I learned. Through continuous learning and practice, by the end of the Signals and Systems course, I have been able to independently complete some simple projects, such as noise analysis, speech synthesis, radar motion detection, etc. I will continue to train to equip myself with more knowledge and skills and be able to solve more problems.  12112301 陈若翀：  For task1, task2, task3, we found that our results were basically consistent with the expected results by comparing with the expected results. Through this project, we learned how to use Doppler effect to check the position and speed of target objects, how to use cross-correlation formula to calculate related values, and learned new knowledge of MATLAB. This experiment allows us to better apply our knowledge of signals and systems to real life problems. In the project, we once found that the traditional method would consume a lot of time when drawing the time Doppler spectrum, and then improved the calculation speed greatly by using the fast Fourier transform. This also allows us to have different understanding and thinking of different algorithms, which is conducive to our next learning.  12011810 王泽奇：  Through this experiment, I have a better understanding of the application of signal technology in real life, and have a certain understanding of cutting-edge signal technology. In addition, the fuzzy function in this experiment is a very novel mathematical method for me, and this way of thinking may play a certain role in my future experimental design. In this experiment, I also fully understood the simulation process of motion signal detection  小组成员分工：  闵泓硕：introduction，report整合  陈若翀：task2, task 3  聂宇康：task1  王泽奇：task3 | |
| **Score** | 闵泓硕：93  陈若翀：87  聂宇康：87  王泽奇：85 |

**Code:**

**Task 1.**

clear

clc

clf

sampleshift=(0:1:5);%时移范围

dopf=(-40:2:40);%多普勒频移范围

fd=-3e6;%频移

bandwidth = 9e6;%带宽

f\_c=2.213e9;%载波频率

f\_s=25e6;%采样率

l=3e8/f\_c;%波长

range=(sampleshift/f\_s)\*3e8;%距离范围

fcut=bandwidth;%截止频率

[b,a]=butter(20,fcut/(f\_s/2));

load("data\_1.mat");

%验证低通滤波器

figure(1);

[h1,f1]=freqz(b,a,512,f\_s);

plot(f1,abs(h1))

xlabel('Frequency (Hz) ');

ylabel('amplitude');

%数字下变频

seq\_ref\_ddc=seq\_ref.\*exp(-1i\*2\*pi\*fd\*[0:duration\*f\_s-1]/f\_s);

seq\_sur\_ddc=seq\_sur.\*exp(-1i\*2\*pi\*fd\*[0:duration\*f\_s-1]/f\_s);

%lpf

seq\_ref\_lpf = filter(b,a,seq\_ref\_ddc);

seq\_sur\_lpf = filter(b,a,seq\_sur\_ddc);

% waveform and spectrum

duration\_p = 0.01;%取前10ms数据绘制时域及频域图

t\_axis = 0 : 1/f\_s : duration\_p-1/f\_s;

f\_axis = -f\_s/2 : f\_s/(duration\_p\*f\_s-1) : f\_s/2;%计算绘图用坐标

ref\_raw\_plot=seq\_ref(1,1:duration\_p\*f\_s);%原始参考信号

sur\_raw\_plot=seq\_sur(1,1:duration\_p\*f\_s);%原始监视信号

ref\_ddc\_plot=seq\_ref\_ddc(1,1:duration\_p\*f\_s);%DDC参考信号

sur\_ddc\_plot=seq\_sur\_ddc(1,1:duration\_p\*f\_s);%DDC监视信号

ref\_lpf\_plot=seq\_ref\_lpf(1,1:duration\_p\*f\_s);%LPF参考信号

sur\_lpf\_plot=seq\_sur\_lpf(1,1:duration\_p\*f\_s);%LPF监视信号

f\_seq\_ref\_lpf\_plot=fftshift(fft(ref\_lpf\_plot));

f\_seq\_ref\_ddc\_plot=fftshift(fft(ref\_ddc\_plot));

f\_seq\_ref\_raw\_plot=fftshift(fft(ref\_raw\_plot));%参考信号时域转频域

f\_seq\_sur\_lpf\_plot=fftshift(fft(sur\_lpf\_plot));

f\_seq\_sur\_ddc\_plot=fftshift(fft(sur\_ddc\_plot));

f\_seq\_sur\_raw\_plot=fftshift(fft(ref\_raw\_plot));%监视信号时域转频域

figure(2)

subplot(3,2,5)

plot(t\_axis\*1e3,real(ref\_lpf\_plot));

xlabel('Time(ms)'),ylabel('Amplitude'),title('Reference signal(after LPF)');

subplot(3,2,6)

plot(f\_axis/1e6,20\*log10(abs(f\_seq\_ref\_lpf\_plot)));

xlabel('Frequency(MHz)'),ylabel('Amplitude(dB)'),title('Spectrum(after LPF)');

subplot(3,2,3)

plot(t\_axis\*1e3,real(ref\_ddc\_plot));

xlabel('Time(ms)'),ylabel('Amplitude'),title('Reference signal(after DDC)');

subplot(3,2,4)

plot(f\_axis/1e6,20\*log10(abs(f\_seq\_ref\_ddc\_plot)));

xlabel('Frequency(MHz)'),ylabel('Amplitude(dB)'),title('Spectrum(after DDC)');

subplot(3,2,1)

plot(t\_axis\*1e3,real(ref\_raw\_plot));

xlabel('Time(ms)'),ylabel('Amplitude'),title('Reference signal(raw)');

subplot(3,2,2)

plot(f\_axis/1e6,20\*log10(abs(f\_seq\_ref\_raw\_plot)));

xlabel('Frequency(MHz)'),ylabel('Amplitude(dB)'),title('Spectrum(raw)');

figure(3)

subplot(3,2,5)

plot(t\_axis\*1e3,real(sur\_lpf\_plot));

xlabel('Time(ms)'),ylabel('Amplitude'),title('Surveillance signal(after LPF)');

subplot(3,2,6)

plot(f\_axis/1e6,20\*log10(abs(f\_seq\_sur\_lpf\_plot)));

xlabel('Frequency(MHz)'),ylabel('Amplitude(dB)'),title('Spectrum(after LPF)');

subplot(3,2,3)

plot(t\_axis\*1e3,real(sur\_ddc\_plot));

xlabel('Time(ms)'),ylabel('Amplitude'),title('Surveillance signal(after DDC)');

subplot(3,2,4)

plot(f\_axis/1e6,20\*log10(abs(f\_seq\_sur\_ddc\_plot)));

xlabel('Frequency(MHz)'),ylabel('Amplitude(dB)'),title('Spectrum(after DDC)');

subplot(3,2,1)

plot(t\_axis\*1e3,real(ref\_raw\_plot));

xlabel('Time(ms)'),ylabel('Amplitude'),title('Surveillance signal(raw)');

subplot(3,2,2)

plot(f\_axis/1e6,20\*log10(abs(f\_seq\_sur\_raw\_plot)));

xlabel('Frequency(MHz)'),ylabel('Amplitude(dB)'),title('Spectrum(raw)');

**Task 2 & Task 3.**

clc;

close all;

addpath('data');

time=1; % figure的编号

for startTime = [1 5 11 18]

load(sprintf('data/data\_%d.mat', startTime));

CIT = 0.5;

fddc = 2.5e6;

bandwidth = 1e7;

t = 0: CIT\*f\_s-1;

% 与task1相同, 原始数据 --> DDC --> 低通滤波

seq\_refddc = seq\_ref.\*exp(1i\*2\*pi\*fddc\*t/f\_s);

seq\_surddc = seq\_sur.\*exp(1i\*2\*pi\*fddc\*t/f\_s);

[b, a] = butter(20, bandwidth/(f\_s/2));

seq\_reflpf = filter(b, a, seq\_refddc);

seq\_surlpf = filter(b, a, seq\_surddc);

DopplerFrequency = -40: 2: 40; % 设置需遍历的频移值

sampleShift = 0: 1: 6;

N = length(seq\_ref);

f\_axis = linspace(-f\_s/2, f\_s/2, N);

T = length(sampleShift);

temp = zeros(T, N);

for t=1: T

% surtime = [seq\_surlpf(1: N-sampleShift(t)), zeros(1, sampleShift(t))];

% reftime = [seq\_reflpf(sampleShift(t)+1: end), zeros(1, sampleShift(t))];

surtime=seq\_surlpf;

tau=t-1; % t要减1试试

reftime=circshift(seq\_reflpf,tau); % cirshift将数组循环平移tau个位置

temp(t,:)=fftshift(fft(surtime.\*conj(reftime))); % fftshift将零频分量移到频谱中心

end

% 以下为作图

array\_range = sampleShift/f\_s\*3e8;

temp\_plot = temp(: , (-19+N/2: 21+N/2));

%归一化

for s=1:7

for k=1:41

temp\_plot(s,k)=abs(temp\_plot(s,k));

if(temp\_plot(s,k)<5e-3)

temp\_plot(s,k)=1e-3;

end

end

end

fig = figure(time);

time = time+1;

[X,Y] = meshgrid(DopplerFrequency, array\_range);

surf(X, Y, abs(temp\_plot));

view(0, 90);

colorbar;

xlabel('Doppler frequency(Hz)');

ylabel('Range(m)');

ylim([0 72]);

set(gca, 'yTick', 0: 12: 72);

set(gca, 'yTickLabel', {'0', '12', '24', '36', '48', '60', '72'});

temp = sprintf('Range-Doppler Spectrum, start time:%5.2f', 0.5\*startTime-0.5);

title(temp)

end

Task3

clc;

close all;

addpath('data');

time = 1;

for startTime = 1: 20

load(sprintf('data/data\_%d.mat', startTime));

CIT = 0.5;

fddc = 2.5e6;

bandwidth = 1e7;

t = 0: CIT\*f\_s-1;

seq\_refddc = seq\_ref.\*exp(1i\*2\*pi\*fddc\*t/f\_s);

seq\_surddc = seq\_sur.\*exp(1i\*2\*pi\*fddc\*t/f\_s);

[b, a] = butter(20, bandwidth/(f\_s/2));

seq\_reflpf = filter(b, a, seq\_refddc);

seq\_surlpf = filter(b, a, seq\_surddc);

DopplerFrequency = -40: 2: 40;

lamda = 3e8/f\_c;

es = 0.07;

d = es/lamda;

sampleShift = 0: 1: 19;

N = length(seq\_ref);

f\_axis = linspace(-f\_s/2, f\_s/2, N);

T = length(sampleShift);

temp = zeros(T, N);

% 以上除startTime和sampleShift外均与task2相同

for t = 1: T

surtime = seq\_surlpf;

tau = t - 1;

reftime = circshift(seq\_reflpf, tau);

temp(t,: ) = fftshift(fft(surtime.\*conj(reftime)));

end

array\_range = sampleShift/f\_s\*3e8;

temp\_plot = temp(: , (-19+N/2: 21+N/2));

time\_plot(time, : ) = max(temp\_plot); % 依次储存1~20的结果

[X,Y] = meshgrid(DopplerFrequency, array\_range);

time = time+1;

end

% 以下为作图

surf(X, Y, abs(time\_plot));

view(0, 90);

colorbar;

xlabel('Doppler frequency(Hz)');

ylabel('Time(s)');

ylim([0 220]);

set(gca, 'yTick', 0: 12: 252);

set(gca, 'yTickLabel', {'0', '0.5', '1', '1.5', '2', '2.5', '3', '3.5', '4', '4.5', '5', '5.5', '6', '6.5', '7', '7.5', '8', '8.5', '9', '9.5'});

temp = sprintf('Time-Doppler Spectrum');

title(temp);